



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MOBILNÍ A WEBOVÁ APLIKACE PRO PODPORU SKUPINOVÉ PRÁCE**

MOBILE AND WEB APP FOR SUPPORTING GROUP WORK

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. PETR POLANSKÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**prof. Ing. ADAM HEROUT, Ph.D.**

**BRNO 2018**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

**Zadání diplomové práce**

Řešitel: **Polanský Petr, Bc.**

Obor: Management a informační technologie

Téma: **Mobilní a webová aplikace pro podporu skupinové práce**

**Mobile and Web App for Supporting Group Work**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Seznamte se s problematikou návrhu a vývoje mobilních aplikací; zaměřte se na platformu Android.
2. Seznamte se s problematikou tvorby webových aplikací a API pro komunikaci s mobilní aplikací.
3. Vyhledejte a analyzujte existující aplikace řešící obdobný problém.
4. Navrhněte a prototypujte způsob interakce s aplikací a jednotlivé prvky uživatelského rozhraní.
5. Navrhněte a implementujte řešenou mobilní aplikaci.
6. Navrhněte a implementujte serverovou část a webové uživatelské rozhraní.
7. Testujte vytvořené aplikace na uživateli a iterativně je vylepšujte.
8. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování; vytvořte plakátek a krátké video pro prezentování projektu.

**Literatura:**

- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN-13: 978-0321965516
- Android Developers: <https://developer.android.com/index.html>
- Susan M. Weinschenk: 100 věcí, které by měl každý designér vědět o lidech, Computer Press, Brno 2012
- Jan Řezáč: Web ostrý jako břitva, Baroque Partners, 2014

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 4, značné rozpracování bodů 5 a 6.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, prof. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
L.S. 612 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato diplomová práce se zabývá návrhem a implementací mobilní a webové aplikace sloužící pro podporu práce v týmu. Jednotliví členové v pravidelných intervalech posílají reporty o své provedené práci pro porovnání se s ostatními. V první části práce je popsána analýza problému a motivace pro vytvoření takové aplikace. Dále jsou popsány a zhodnoceny podobné aplikace, podporující týmovou práci, platforma Android a všechny technologie potřebné pro vývoj. V návrhu jsou popsány jednotlivé obrazovky mobilní aplikace, historie jejich návrhu a návrh webové aplikace. Nakonec je popsána implementace a testování.

## Abstract

This master's thesis describe design and implementation of mobile and web application for supporting group work. Every team member send his work report in specific time period for compare themself with each other. In first part is described analysis and motivation for this application. Next chapters inform about similiar applications, Android platform and used technologies. In design chapter are described screens of mobile application, history of their design and web application design. In the last chapter is described implementation and testing.

## Klíčová slova

Firebase, Android, tým, týmová práce, databáze, úložiště, aplikace, mobil, web, Material Design, Android Studio

## Keywords

Firebase, Android, team, teamwork, database, storage, mobile, application, web, Material Design, Android Studio

## Citace

POLANSKÝ, Petr. *Mobilní a webová aplikace pro podporu skupinové práce*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

# Mobilní a webová aplikace pro podporu skupinové práce

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana prof. Ing. Adama Herouta Ph.D.. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Petr Polanský  
21. května 2018

## Poděkování

Rád bych poděkoval svému vedoucímu panu Prof. Ing. Adamu Heroutovi Ph.D. za jeho velkou trpělivost, ochotu a vedení mé diplomové práce tím správným směrem. Dále bych chtěl poděkovat kolegům Bc. Michalu Řepkovi a Bc. Zdenku Brandejsovi, kteří mi pomohli při návrhu aplikace.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Motivace pro tvorbu aplikace</b>	<b>4</b>
2.1	Nástin řešení . . . . .	4
<b>3</b>	<b>Podobné aplikace pro práci v týmu</b>	<b>6</b>
3.1	Workboard . . . . .	6
3.2	Trello . . . . .	7
3.3	Asana . . . . .	8
3.4	Wrike . . . . .	9
3.5	Komunikační aplikace . . . . .	10
3.6	Zhodnocení aplikací . . . . .	10
<b>4</b>	<b>Informace k vývoji pro mobilní systém Android</b>	<b>12</b>
4.1	Platforma Android . . . . .	12
4.2	Firebase . . . . .	14
4.3	Material Design . . . . .	18
4.4	Poznatky o tvorbě designu aplikací . . . . .	19
4.5	Interakce webové a mobilní aplikace . . . . .	20
4.6	Google Play konzole . . . . .	20
4.7	Zveřejnění aplikace na Google Play . . . . .	21
<b>5</b>	<b>Návrh aplikace NudgeMe</b>	<b>23</b>
5.1	Popis rozložení obrazovek mobilní aplikace . . . . .	23
5.2	Webová aplikace . . . . .	29
5.3	Strukturování dat v databázi . . . . .	30
5.4	Návrhy budoucí práce . . . . .	31
<b>6</b>	<b>Implementace</b>	<b>33</b>
6.1	Popis algoritmu pro vykreslení intenzity práce . . . . .	33
6.2	Mobilní aplikace . . . . .	34
6.3	Webová aplikace . . . . .	38
<b>7</b>	<b>Testování</b>	<b>39</b>
7.1	Využití aplikace . . . . .	40
7.2	Poznatky vyplývající z testování . . . . .	41
7.3	Zhodnocení testování . . . . .	42
<b>8</b>	<b>Závěr</b>	<b>44</b>

<b>Literatura</b>	<b>45</b>
<b>A Obsah přiloženého paměťového média</b>	<b>47</b>

# Kapitola 1

## Úvod

Diplomová práce se zabývá popisem mobilní a webové aplikace, sloužící pro podporu týmové práce. Mým cílem bylo vytvořit aplikaci, která informuje o práci jednotlivých členů týmu a pomůže zlepšit její efektivitu a intenzitu. V úvodní kapitole se věnuji analýze problému a popisu motivace, proč jsem se rozhodl tuto aplikaci vytvořit. Popisuji základní princip programu, jakým způsobem se snažím řešit nastíněný problém a na jaké potenciální uživatele cílím. Dále analyzuji podobná řešení, zabývající se skupinovou prací. Každé z nich hodnotím podle toho, na kolik je podle mého názoru užitečné a čím se liší od mého řešení. Následuje popis nezbytných poznatků pro základní představu o vývoji mobilní aplikace pro Android a pro detailní pochopení zbytku mojí práce. Dále se zabývám obecnými doporučeními, která se týkají designu a přívětivosti aplikace pro uživatele. Poslední část kapitoly jsem věnoval interakci webové a mobilní aplikace, popisu jaké jsem se rozhodl použít řešení a jaké jsou jiné možnosti.

V kapitole 5 informuji o návrhu aplikace, popisuji a zdůvodňuji, ovládání aplikace, rozložení jednotlivých obrazovek, jejich vzhled a své nápady na zlepšení aplikace jako celku. Také porovnávám a vyhodnocuji různé nápady, které mě v průběhu vývoje napadly a zdůvodňuji také, proč jsem od některých z nich nakonec upustil. V implementaci rozebírám nástroje a algoritmy, nad kterými aplikace funguje a jakým způsobem jsem implementoval věci z návrhu. Dále popisuji testování na koncových uživateli, jejich zpětnou vazbu, jakého vytížení aplikace dosahuje a celkově hodnotím získané informace o aplikaci.

V závěru se ohlížím za celou práci, hodnotím její užitečnost, co všechno se stihlo a nestihlo a přikládám představu o tom, jak by se aplikace mohla vyvíjet do budoucna.

## Kapitola 2

# Motivace pro tvorbu aplikace

Spolupráce lidí na jednom úkolu není vždy jednoduchá a mnohdy závisí na prostředí, ve kterém je vykonávána. Pravděpodobně je motivace zaměstnanců ve firmě vyšší než studentů u školního projektu. V práci jsou lidé motivováni efektivnějšími prostředky, jako jsou peníze nebo zlepšení pracovního postavení. Ve škole jsou však studenti motivováni známkou z projektu, někdy je práce baví a mnohdy jen chtějí splnit projekt. Dochází tak k častému odkládání projektů nebo špatné informovanosti členů týmu, takže ve výsledku se na projektu pracuje do posledních chvil před odevzdáním a konečný výsledek neodpovídá očekáváním. Pokud členové týmu mohou používat aplikace pro sdílení obsahu, jako jsou např. *Google dokumenty*, je větší přehled o odvedené práci. Existují však situace a projekty, při kterých tyto nástroje nelze použít, jako jsou manuální práce nebo programování.

V současné době existuje celá řada programů, které slouží pro řízení a plánování projektů, z nichž některé popisuji v následující kapitole. Obvykle mezi ně patří jednoduché aplikace pro tvorbu *seznamu úkolů* (To-do list) nebo robustní aplikace pro velké firmy. Domnívám se však, že s velkými aplikacemi, které jsou mnohdy placené, se nebudou chtít studenti učit pracovat, protože tyto aplikace jsou mnohdy složité a studenti většinu funkcí nevyužijí. Aplikace pro tvorbu seznamů úkolů jsou zase příliš obecné, mnohdy neukazují úspěšnost splnění úkolů a není možné porovnat jednotlivé členy podle odvedené práce.

### 2.1 Nástin řešení

Jedním ze způsobů jak motivovat lidi k práci nebo k lepším výkonům je jejich porovnávání s ostatními. Jak je uvedeno v článku [10], existují různé důvody, proč tato motivace na lidi funguje. Někteří mají soutěživou povahu a jakékoliv poměrování chtějí vyhrát, jiní zase berou konkurenci jako prostředek k tomu, aby dosáhli co nejlepšího výsledku. Právě na motivaci skrze porovnání člena týmu s ostatními jsem se rozhodl vytvořit svoji aplikaci. Práce jednotlivých členů bude vykreslena do spojnicového grafu, ze kterého bude možné vyčíst, jak se liší intenzita práce v rámci týmu. Díky němu lze rychleji zjistit, kdo provádí svoji práci svědomitě a kdo naopak zaostává. Graf bude vykreslen na hlavní obrazovce, která bude zobrazena vždy při zapnutí aplikace. Ta nemá za cíl, aby ji uživatelé používali dlouho, ale spíše krátce a často. Ihned po shlédnutí grafu, bude mít uživatel přehled o tom, jak si kdo vede, kdo v aktuální den zadal nebo nezadal report a na základě toho může vyvodit patřičný závěr. Data do grafu chci získat pomocí pravidelných krátkých reportů, které budou uživatelé posílat, a ve kterých budou ostatní informovat o své práci a také sami sebe hodnotit. Jak je možné se dočíst v knize *Konec prokrastinace* [12], často když



má člověk výběr z více možností, může se u něj projevit rozhodovací paralýza. Ta spočívá v tom, že čím více možností mají lidé na výběr, tím je pro ně těžší se pro některou z nich rozhodnout. Z tohoto důvodu jsem se rozhodl, že uživatel se bude moci ohodnotit pouze třemi známkami, které budou symbolizovat úspěch, mírný úspěch a neúspěch.

Pokud uživatel bude mít díky svým hodnocením úžasné výsledky v grafu, budou moci ostatní tato hodnocení ověřit podle textu, který vyplnil v reportu. Veškeré údaje o reportech každého člena týmu bude možné zobrazit. Text bude povinné zadat a tedy z jeho struktury by mohlo jít identifikovat, zda práci opravdu vykonal. Informováním o práci pomocí obyčejné zprávy, např. přes *Facebook*, může uživatel sdělit, že na zadané části projektu pracuje. Pokud se vedoucí pravidelně nedotazuje na stav dokončení, nemusí zjistit, jak ve skutečnosti práce probíhá. V mé aplikaci by se tato situace mohla částečně eliminovat tím, že každý se ke svoji práci musí pravidelně vyjadřovat. Může pravidelně psát neurčitou odpověď, že na úkolu pracuje, ale pokud by takto postupoval až příliš často, teprve pak by ostatní museli pomocí osobních zpráv vyzvat uživatele k podrobnějším informacím o dokončenosti. Z toho vyplývá moje snaha o zjednodušení a zefektivnění komunikace na projektech a zlepšení přehlednosti o práci druhých.

Pro větší záruku, že uživatelé budou pravidelně posílat reporty, bude sloužit funkce *NudgeMe* ("štouchni mě"). Díky ní, jak již z jejího názvu vyplývá, uživatelé budou moci poslat zprávy (*Nudgee*) s žádostí o vyplnění reportu ostatním členům týmu, kteří doposud report nevyplnili. Vyzývání k zadání reportu pomocí funkce *NudgeMe* by mělo být lepším prostředkem pro vyzvání k zadání reportu než automatické upozornění, které by generovala aplikace. Takto se jedná o zprávu kolegy/kamaráda, který vás o něco žádá, čímž se ze zprávy stává osobnější záležitost. Zároveň vyplňující se tímto způsobem může pochlubit svoji prací. Zpráva zase není tak osobní, že by se uživatel bál takto někoho upomenout. Kromě případů, kdy uživatel ignoruje veškeré pokusy donutit jej k nějaké aktivitě, může právě *Nudge* nahradit osobní zprávy na *Messengeru*. Stisknutím jednoho tlačítka se osloví všichni uživatelé, kteří doposud report nezadali. Zpráva bude mít již předdefinovaný text a tedy nebude nutné se zdržovat s jeho vymýšlením. Funkce *NudgeMe* nemusí mít jen otravný charakter, kdy uživatel ostatní do něčeho nutí, ale jejím prostřednictvím sděluje, že on už svoji práci odvedl a tím může motivovat ostatní k dokončení svých úkolů.

Pro ještě lepší zobrazení dat bude sloužit webová aplikace. Ta má za cíl vykreslit grafy jednotlivých projektů uživatele vedle sebe, díky čemuž může lépe porovnat, jak se liší intenzita práce na projektech. Dalším cílem je možnost graf zvětšit a lépe tak zobrazit data reportů. Je pochopitelné, že pokud uživatel pracuje na více projektech, nebude úroveň intenzity práce na každém projektu stejná. Někdy bude pracovat na jednom, pak na druhém. K tomu právě slouží prostřední hodnota, kterou se může uživatel ohodnotit, sloužící pro vysvětlení, že na projektu nemohl pracovat, protože dělal jinou důležitou věc.

Hlavní cílovou skupinou jsou studenti vysokých škol, kteří řeší skupinové projekty. Jsem toho názoru, že tito mladí lidé mají větší touhu si něco dokazovat a soupeřit s ostatními. Také jsou otevření novým věcem, a zvláště pokud se jedná o nástroj pro zefektivnění jejich práce. Zacílení právě na tuto skupinu je také z toho důvodu, že na středních školách se týmové projekty příliš nezadávají a lidé, kteří pracují na týmových projektech v zaměstnání zase mohou mít pokročilejší nástroje s podrobnějšími ukazateli a statistikami. Aplikace však není nutné použít pouze pro řešení týmového projektu, ale i jako nástroj, díky kterému se lidé mohou porovnávat mezi sebou při provádění nějaké činnosti. Stačí, aby si uživatelé zvolili nějaký společný cíl nebo činnost a do reportů sdělovali informace o jejich plnění.

Cílem této práce pak je zjistit, zda tento nástroj má mezi uživateli uplatnění a zda dokáže zlepšit práci v týmu.

## Kapitola 3

# Podobné aplikace pro práci v týmu

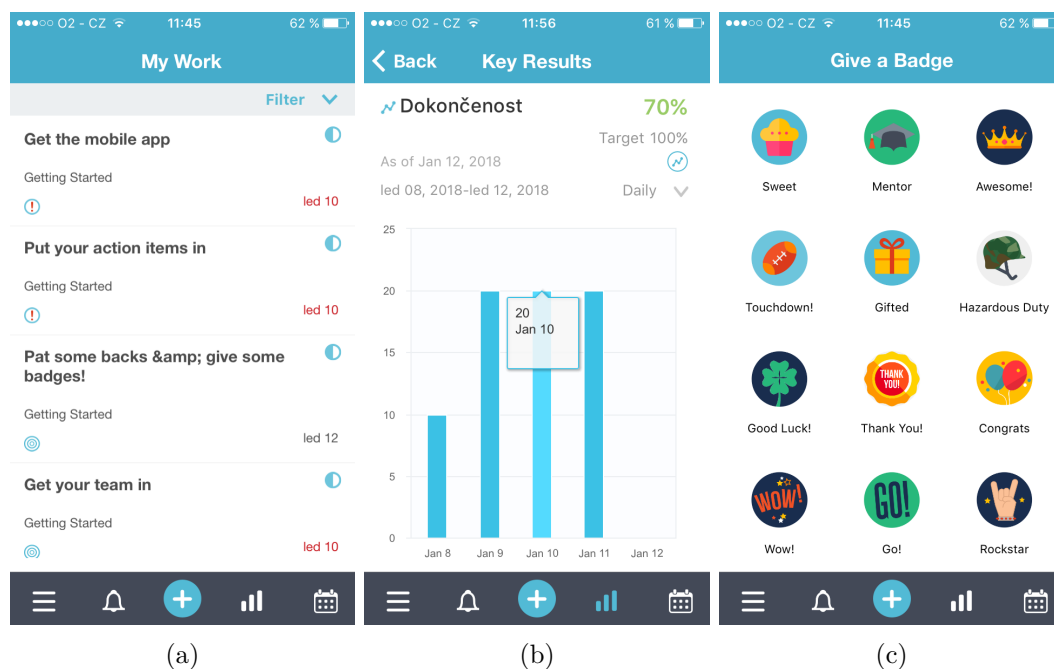
V této kapitole jsou popsány typy aplikací, které úzce souvisí s týmovou prací a komunikací mezi členy týmu. Každá z nich zastupuje jiný přístup řešení dané problematiky. Popisuji je z toho důvodu, abych lépe vysvětlil a odůvodnil vznik svojí aplikace.

### 3.1 Workboard

Tato aplikace<sup>1</sup> je podobná té, kterou vytvářím, ale informování o provedené práci u ní probíhá trochu odlišně. Hlavní položkou je seznam s úkoly, jak je uvedeno na obrázku 3.1, který slouží i pro rozdělování práce mezi spolupracovníky. Je možné stanovit určitý cíl projektu, kterého chce uživatel dosáhnout a každý den vyplňovat informace, nakolik se mu povedlo k tomuto cíli přiblížit. Není zde ale možné graficky znázornit výkon celého týmu do jednoho grafu. Také je stanovování cílů spíše bráno jako dodatečná funkcionality, než jako klíčový ukazatel. Stejně jako moje aplikace umožňuje vyplnit report jen od začátku projektu do aktuálního dne. Jedná se tedy o aplikaci, která se svojí funkcionalitou nachází někde mezi aplikacemi vytvářející seznam úkolů a profesionálním vývojovým softwarem.

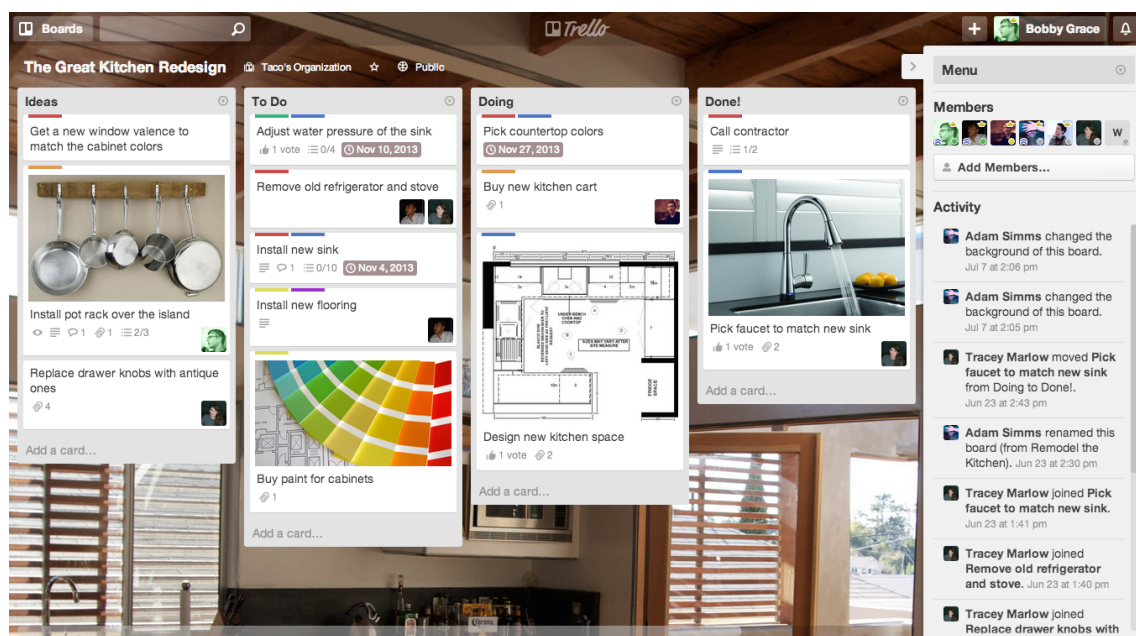
---

<sup>1</sup>Workboard — <https://www.workboard.com/>



Obrázek 3.1: Ukázka zobrazuje nejprve seznam úkolů 3.1a pro konkrétního člena týmu. Dále je zobrazení pokroku na určitém úkolu 3.1b, kde parametr Dokončenost udává 70% a sloupcový graf říká, kolik procent práce uživatel udělal v konkrétní den. Za určitou aktivitu je možné udělit ostatním členům týmu odznak, jejichž výčet je na poslední obrazovce 3.1c.

## 3.2 Trello



Obrázek 3.2: Pohled na nástěnku s úkoly a sloupci, popisující určitý stav dokončenosti. Pokud se stav některých úkolů změnil, uživatel jej jednoduše metodou drag & drop přesune do jiného sloupce. K úkolům jsou přiřazeni určití členové, obrázky, ankety nebo seznamy.

Další aplikace, kterou jsem se inspiroval, pracuje s kartičkami úkolů a jejich řazením do patřičných sloupců, definujících stav úkolu, podle kterých spolupracovníci poznají, v jaké fázi dokončení se daný úkol nachází. *Trello*<sup>2</sup> jsem zvolil z důvodu jeho velké popularity a také proto, že patří mezi nejjednodušší aplikace používající kartičky pro definování pracovního postupu. Jak je ukázáno na obrázku 3.2, nástěnka je přehledná a dává důraz na co nejrychlejší vytvoření nového úkolu. Při kliknutí na konkrétní úkol se však nabízí více funkcionalit, jako je vložení obrázku, přidání přílohy, hlasování nebo stanovení termínu pro dokončení.

Inspirací pro moji aplikaci mi byl způsob, jakým mezi sebou jednotliví spolupracovníci komunikují. Při pohledu na nástěnku se dá zjistit, na čem daný člověk pracuje, zda daná práce neměla už být hotová dříve a kolik práce mu ještě zbývá udělat. Z mého pohledu však nenutí uživatele práci rozdělit na menší části, které by dávaly více informací jeho kolegům. Pracovník tedy může mít jednu kartičku s obecným úkolem ve stejné fázi rozpracování i několik dní a ostatní členové nebudou vědět, zda práce probíhá podle plánu nebo se na něčem zasekl. Pro tyto případy je možné přidávat ke kartě komentáře a ptát se uživatele na detailnější informace o stavu jeho práce, avšak tím pro ostatní vyvstává povinnost se aktivně informovat. V případě, že uživatel neprojeví aktivitu a nezeptá se, může se stát, že úkol přesáhne konečný termín pro dokončení.

### 3.3 Asana

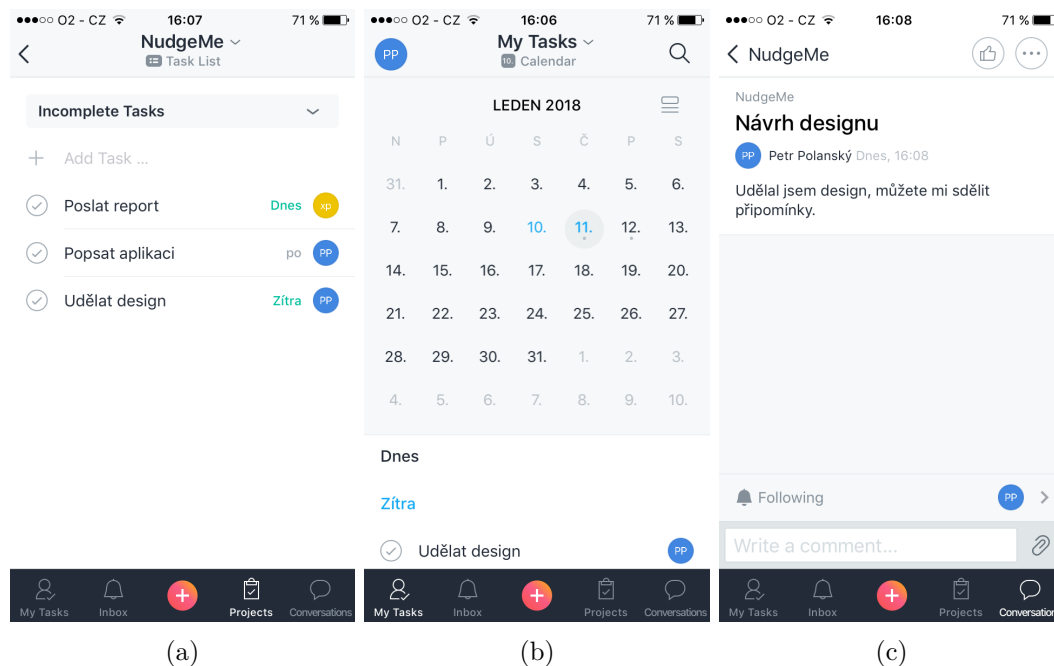
Jedná se o aplikaci pro tvorbu seznamu úkolů k provedení, jejíž ukázky jsou na obrázku 3.3. Podobně jako Trello jsou zde vytvářeny úkoly, které mohou být přiřazeny určitému uživateli (viz. obrázek 3.3a). Uživatelé konkrétního projektu také mohou vést konverzaci, jak je ukázáno na obrázku 3.3c, a posílat si soubory. K dispozici je také nástěnka, která obsahuje graf projektu. Ten informuje o tom, kolik je ke konkrétnímu dni splněno úkolů a kolik jich ještě zbývá.

*Asana*<sup>3</sup> bohužel neumožňuje specifikovat stav úkolu jinak, než na neuděláno/hotovo a tím pádem je informovanost členů týmu ještě horší než v případě Trelly. Graf stavu práce zase neudává, jestli v průběhu plnění úkolů nenastaly nějaké komplikace, které se mohou znovu objevit.

---

<sup>2</sup>Trello — <https://trello.com/>, obrázek byl převzat z této stránky

<sup>3</sup>Asana — <https://asana.com/>



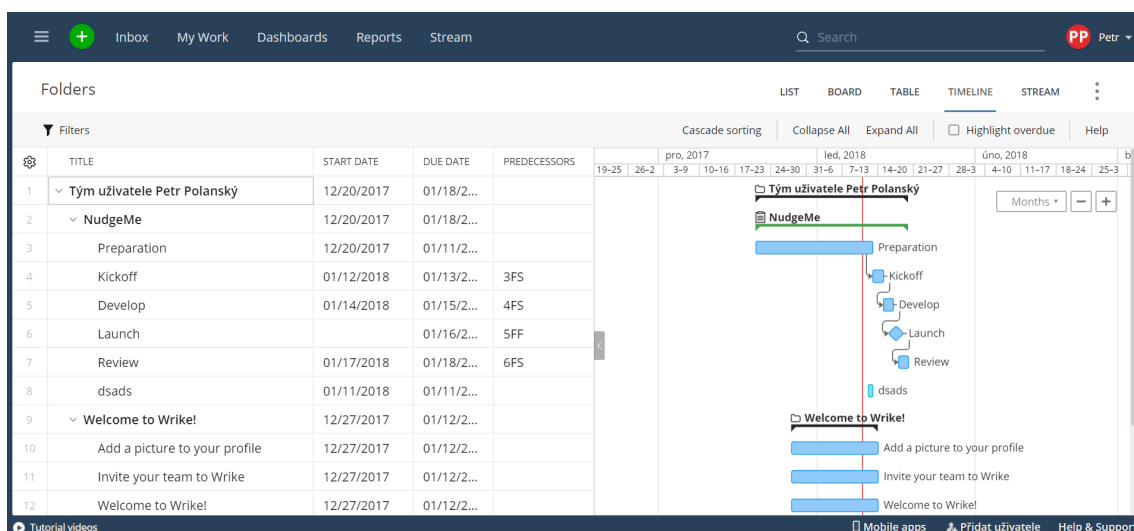
Obrázek 3.3: Soubor obrázků ukazuje aplikaci Asana a její funkce. Zcela nalevo je obrázek seznamu úkolů, u kterých je ve zkratce uveden vlastník. V kalendáři 3.3b jsou pak úkoly zaznačeny podle termínů, do kdy mají být dokončeny. Na poslední obrazovce je ukázka posílání zpráv v rámci daného projektu.

## 3.4 Wrike

Jedná se o program<sup>4</sup> sloužící pro profesionální správu a řízení projektů pro větší i menší podniky. Umožňuje plánovat projekt pomocí *Ganttova diagramu*<sup>5</sup>, který je ukázán na obrázku 3.4, nebo diagramu zobrazujícího vytížení jednotlivých členů týmu. Každý úkol má stanovený začátek a konec a jeho doba trvání je vyobrazena v kalendáři aplikace. Další funkce, jako je vkládání souborů, psaní zpráv a zobrazení nástěnky, jsou podobné jako u Trelly a Asany.

<sup>4</sup>Wrike — <https://www.wrike.com>

<sup>5</sup>Jedná se o pruhový diagram, který znázorňuje posloupnosti úkolů a jejich závislosti v čase.



Obrázek 3.4: Náhled obrazovky pro plánování ve Wrike. Nalevo se nachází seznam jednotlivých úkolů s časovými intervaly pro vykonání. Na pravé straně se nachází Ganttův diagram, který zobrazuje jednotlivé úkoly v pořadí plnění a jejich závislosti mezi sebou.

### 3.5 Komunikační aplikace

Posledním typem aplikací, které slouží pro podporu práce v týmu, jsou aplikace pro psaní zpráv. Mezi nejznámější z nich patří *Messenger*<sup>6</sup> od *Facebooku* nebo *WhatsApp*<sup>7</sup>. Obě jsou velmi rozšířené, protože účet na Facebooku má v dnešní době skoro každý a k přístupu na WhatsApp stačí telefonní číslo, které má ještě více lidí. Tyto dvě aplikace ale slouží spíše pro komunikaci mladých lidí a případně pro probírání prací na menších projektech.

Specializovanější aplikací je pak *Slack*<sup>8</sup>, který umožňuje uživateli vytvářet vlastní kanály komunikace, do kterých pouze pozve svoje spolupracovníky. Zároveň umožňuje propojení s některými dalšími aplikacemi jako je *Google Drive*, *Wrike* nebo *Git*<sup>9</sup>.

V rámci komunikace je možné použít aplikace pro zprostředkování hovorů přes internet jako je *Skype*<sup>10</sup>, *Viber*<sup>11</sup> nebo již zmíněný WhatsApp.

Tyto aplikace však slouží spíše pro diskuzi a výměnu názorů. Oproti tomu moje aplikace má za cíl sdělit jednoduše a stručně spolupracovníkům, jak je na tom každý člen týmu s plněním svých úkolů, které mohly být diskutovány a ustanoveny pomocí výše zmíněných aplikací.

### 3.6 Zhodnocení aplikací

Každá z popsaných aplikací mě určitým způsobem ovlivnila. Workboard umožňuje práci zobrazit pomocí grafu, ale už zde není možné její porovnání s ostatními členy týmu. Zajímavé a inspirující je procentuální stanovení dokončenosti práce a stanovení cíle práce. Tento

<sup>6</sup>Messenger — <https://www.messenger.com>

<sup>7</sup>WhatsApp — <https://www.whatsapp.com>

<sup>8</sup>Slack — <https://slack.com>

<sup>9</sup>Git — systém sloužící pro správu verzí souborů

<sup>10</sup>Skype — <https://www.skype.com>

<sup>11</sup>Viber — <https://www.viber.com>

přístup by v mojí aplikaci mohl uživatele více motivovat, když by měl cíl nebo podúkoly. Trello jsem si spíše vybral, abych poukázal na jeho nedostatky co se týče informovanosti uživatelů. Inspiroval jsem se však jeho schopností na hlavní obrazovce stručně informovat o průběhu práce. Asana mě zaujala svým samostatným chatem, díky čemuž uživatelé nemusejí využívat jiného programu. Na podobném principu jsem udělat funkci NudgeMe, pouze jsem se nesnažil o úplné nahrazení jiných nástrojů, které tuto práci zvládají spolehlivě. Díky jejímu kalendáři, který určitým způsobem informuje o vymezeném intervalu pro projekt, jsem ve svém kalendáři omezil možnost zadat report pouze pro dny, kdy projekt trvá. Wrike jsem zvolil jakožto zástupce velkých aplikací pro podporu celkového managementu firmy, aby bylo možné vidět, jak se liší priority oproti školním projektům. Zde je kladen důraz na zobrazení co nejvíce údajů o plnění úkolů a jejich časovém rozložení. Plánování a kontroly mnohdy neprobíhají denně, ale po delších časových úsecích. Zajímavé na Ganttově diagramu je jeho schopnost informovat, že vlivem nesplnění úkolu do zadaného data, se posune provádění všech dalších prací, které na něj navazují. Lze tak dlouho dopředu například určit, že projekt se zpozdí. Tuto částečnou předvídatelnost budoucnosti by nebylo špatné zakomponovat do aplikace. Z komunikačních aplikací jsem se především inspiroval designem a způsobem provádění některých operací. V případě Messengeru jsem si vypůjčil způsob zobrazení ikon uživatelů na hlavní obrazovce. V mé aplikaci však neslouží pro přechod na konverzaci s konkrétním uživatelem, ale na výpis jeho reportů. Na stejném principu jako funguje Slack jsem navrhl přepínání mezi projekty pomocí postranního menu. Dále mě zaujala možnost tento nástroj propojit s dalšími aplikacemi, což by v mém řešení jistě také našlo uplatnění.

## Kapitola 4

# Informace k vývoji pro mobilní systém Android

V této kapitole popisují důležité věci, které jsou nutné pro pochopení toho, jak funguje platforma Android, jaké technologie jsem se rozhodl použít, a které poznatky jsou důležité pro přizpůsobení aplikace uživateli. Více jsem rozepsal platformu *Firebase*, ze které jsem použil její databázi, úložiště a posílání zpráv. Právě tyto nástroje jsem pak rozvedl do větších podrobností. Podkapitolu o platformě *Android* jsem převzal z dokumentace [1], kromě popisu *Android Studio*.

### 4.1 Platforma Android

Jedná se o operační systém, který vytvořila firma Google. V současnosti je systém dostupný pro mobily, tablety nebo chytré hodinky. Android je *víceuživatelský linuxový systém*, kde každá aplikace vystupuje jako jiný uživatel, má unikátní identifikátor a běží jako vlastní proces, který má svůj vlastní *virtuální stroj*. Podle příručky pro vývojáře [1] jsou základními stavebními kameny Android aplikace tzv. *aplikační komponenty*. Každá z nich má jasně stanovenou funkci a životní cyklus. Dělí se na *Activities*, *Services*, *Broadcast receivers* a *Content providers*.

- **Activities** — slouží pro interakci s uživatelem, každá z nich představuje jedno okno aplikace s definovaným rozhraním, mohou spustit jinou aktivitu, která si uchovává informace o předešlé pro případ, že by se na ni chtěl uživatel vrátit,
- **Services** — zajišťují běh procesů na pozadí aplikace,
- **Broadcast receivers** — zajišťují doručení událostí, které mohou být doručeny i v případě, že aplikace neběží,
- **Content providers** — slouží pro řízení a sdílení aplikačních dat s možností je uložit do souborů nebo databází.

*Activities*, *Services* a *Broadcast receivers* je možné aktivovat asynchronní zprávou zvanou *Intent*. Android funguje na principu nejnižších privilegií, kde každá aplikace má přístup jen k těm zdrojům, které potřebuje. Existují však mechanismy, kdy aplikace může sdílet informace s jinou aplikací. Pro instalaci aplikace na zařízení slouží soubor *APK*<sup>1</sup>, který je

---

<sup>1</sup>APK — Android Package Kit



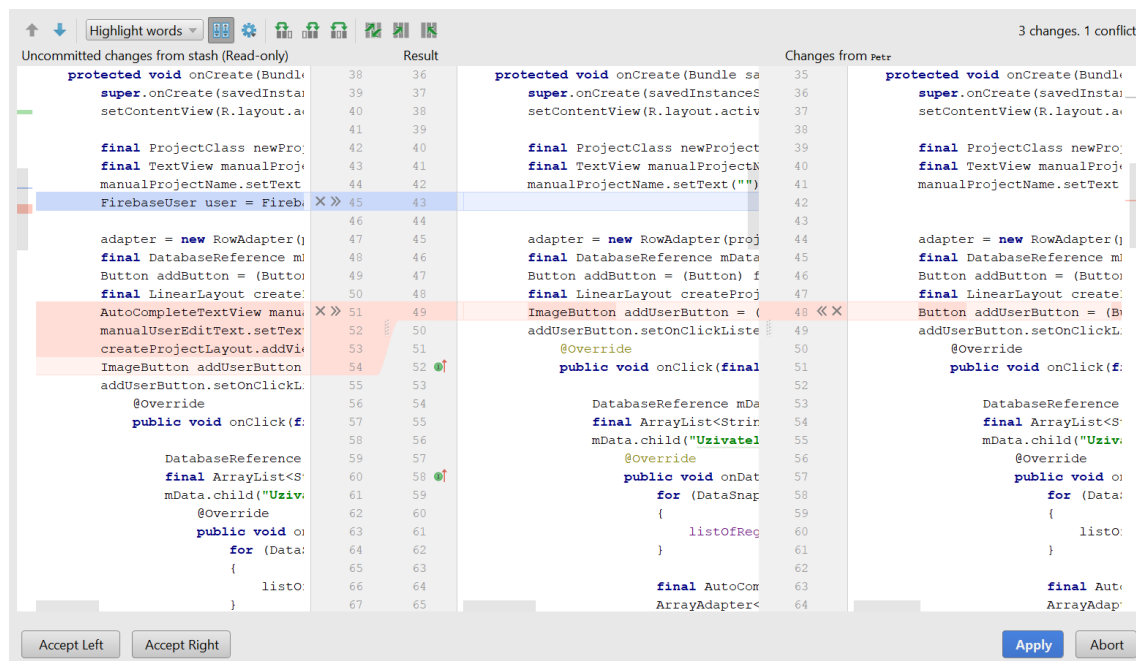
vytvořen zkompilem zdrojových kódů a ostatních souborů, jako jsou aplikační zdroje (resources), slouží pro vizuální prezentaci aplikace. Jedním ze zdrojů jsou *XML*<sup>2</sup> soubory, díky kterým může uživatel definovat styly, barvy nebo animace pro svoji aplikaci. Každý tento zdroj má přiřazen unikátní identifikátor, přes který je možné se na zdroj odkazovat v kódu.

## Android Manifest

Jedná se o soubor, který slouží pro specifikování práv pro aplikaci, API<sup>3</sup> knihoven nebo hardwaru a softwaru, který může aplikace využívat (kamera, bluetooth). Dále se mohou specifikovat požadavky na zařízení a jeho zdroje, aby bylo zaručeno, že aplikace bude fungovat. To slouží např. při znemožnění stažení aplikace z Google Play, pokud zařízení nemá vyhovující prostředky pro běh aplikace. Pokud chce uživatel využívat výše zmíněné aplikační komponenty, musí je v tomto souboru také uvést.

## Android Studio

Jedná se o *IDE*<sup>4</sup> vytvořené firmou Google pro programování aplikací pro platformu Android. Jednou z hlavních předností tohoto prostředí je zabudovaný emulátor mobilních zařízení. Vývojář si může vybrat mezi různými typy zařízení i verzí operačního systému. Nahrávání aplikace do emulátoru je jednoduché a v případě menší změny se aplikace nemusí ani restartovat. Tímto způsobem testování může uživatel dopředu zjistit, jak se jeho aplikace přizpůsobuje určitému rozměru úhlopříčky displaye.



Obrázek 4.1: Nástin řešení konfliktů v Android Studiu. Situace nastává, když uživatel přepíná mezi větvemi při používání Gitu. Na obrázku je vyobrazen konflikt mezi změnami, na které ještě nebyl spuštěn commit, a větví s názvem Petr.

<sup>2</sup>XML — Extensible Markup Language

<sup>3</sup>API — Application Programming Interface

<sup>4</sup>IDE — Integrated Development Environment

Název technologie	Popis
<b>Analytics</b>	sledování výkonnosti a používání aplikace
<b>Authentication</b>	přihlášení přes email, Facebook, Git atd.
<b>Test lab</b>	testování aplikace na zařízeních v datovém centru Googlu
<b>Remote Config</b>	změnění aplikace bez nutnosti vytváření aktualizace
<b>AdMob</b>	řešení pro vložení reklamy do aplikace
<b>Cloud Firestore</b>	alternativa k databázi pracující v reálném čase

Tabulka 4.1: Výčet některých užitečných funkcí, které nabízí Firebase.

Pro podporu tvorby kódu ve více lidech je k dispozici Git. Je vhodný zejména při obecných úkonech, jako je stažení repozitáře a poslání změn na server. Při řešení případných konfliktů lze použít porovnání jednotlivých souborů za pomoci *GUI*<sup>5</sup>, které je ukázáno na obrázku 4.1, kde uživatel vybere, které změny z jakého souboru chce ponechat ve finální verzi. Jedna z doplňujících funkcí je možnost kontroly a odstranění referencí na knihovny, které jsou vloženy v kódu, ale v aplikaci nejsou využity.

Studio používá pro sestavení aplikace nástroj *Gradle*<sup>6</sup>, který zajišťuje jednoduchý import knihoven. Díky tomu dochází k snadnému propojení s knihovnou Firebase. Další užitečnou funkcí je *Vector asset* sloužící k vložení a převodu vektorových obrázků na zdrojový soubor XML. Jsou k dispozici ikony *Material Designu*, který popisují dále v této kapitole.

## 4.2 Firebase

Jedná se o platformu, která slouží pro podporu vývoje jak mobilních, tak i webových aplikací. Původní firma byla koupena Googlem a díky tomu Firebase snáze spolupracuje s ostatními Google funkcemi a nástroji, jako je např. Android Studio. Vývoj je také umožněn pro *iOS*<sup>7</sup> a *Unity*<sup>8</sup>. Ve své aplikaci jsem použil databázi pracující v reálném čase, funkce pro autentizaci, úložiště a posílání zpráv přes cloud. Všechny tyto technologie jsem si dovolil popsat v následujících podkapitolách. Tabulka 4.1 ve stručnosti popisuje další užitečné funkce Firebase, které jsou pak detailněji popsány v příslušné dokumentaci [4].

### Firebase konzole

Tato konzole<sup>9</sup> slouží pro nastavení Firebase pro uživatele a zároveň shromažďuje některé statistiky, jak je možné vidět na obrázku 4.2. Statistiky se týkají hlavně toho, jak jsou v aplikaci využívány Firebase nástroje. Pro autentizaci je zde možné povolit či zakázat různé formy přihlašování (email, Facebook, Google atd.) nebo vidět seznam všech registrovaných uživatelů, přičemž jim lze přes konzoli zaslat ztracená hesla, která jsou vývojáři dostupná pouze v zahashované podobě. V rámci konzole je možné mít více projektů, mezi kterými lze snadno přepínat. Také je možné přidat práva ostatním uživatelům pro přístup k projektu nebo poslat na zařízení zprávu přes *Cloud Messaging*, který je popsán dále.

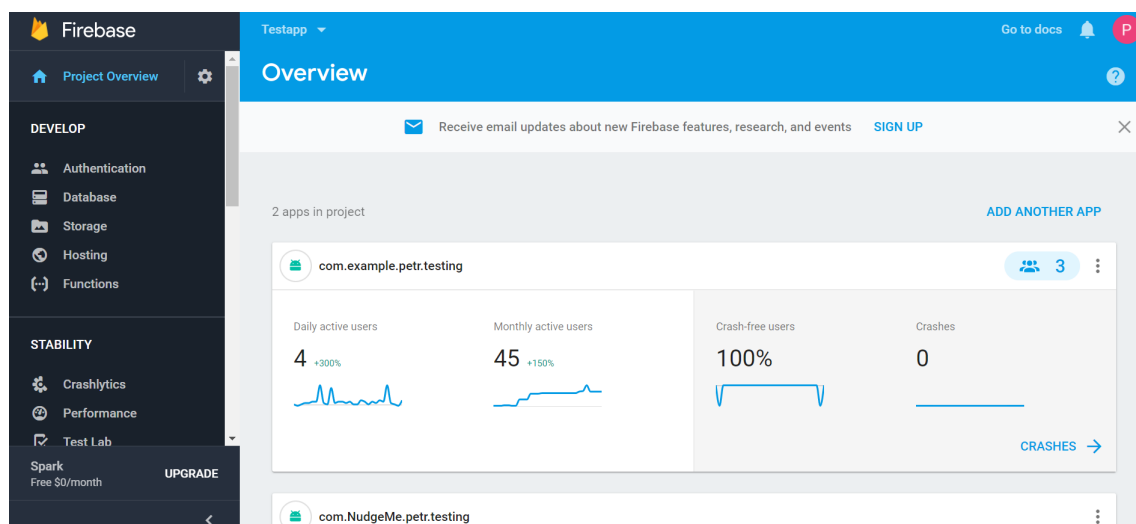
<sup>5</sup>GUI — Graphical User Interface

<sup>6</sup>Gradle — <https://gradle.org>

<sup>7</sup>iOS — jde o mobilní operační systém vyvinutý firmou Apple Inc., který se používá pro Iphony a iPady

<sup>8</sup>Unity — jedná se o herní engine vyvinutý společností Unity Technologies

<sup>9</sup>Firebase konzole — <https://console.firebase.google.com/>



Obrázek 4.2: Jedná se o nástěnku Firebase konzole, kde vlevo je panel s dostupnými nástroji, které lze v konzoli spravovat a uprostřed jsou základní statistiky o využití aplikace.

Při kliknutí na databázi se zobrazí veškerá data, která jsou v ní uložena ve formě *JSON*<sup>10</sup> struktur. Tato data lze také přes konzoli mazat, přidávat nebo měnit. Databázi lze také specifikovat přístupová práva nebo lze zjistit, kolik databáze zabírá dat nebo jak časté jsou transakce. Stejně funkce jsou pak k dispozici i pro úložiště. Kromě těchto nástrojů, které jsou pro vývoj, je možné konfigurovat další pro kontrolu stability aplikace (sledování výkonu, *TestLab*), pro analýzu (událostí, uživatelů) nebo pro přidání reklamy.

## Databáze pracující v reálném čase

Jedna z hlavních funkcí Firebase je poskytování databáze, která pracuje v reálném čase. Jak už bylo zmíněno, data nejsou uspořádána relačně, ale do JSON struktur a jsou k dispozici na všech zařízeních, na které se uživatel přihlásí. Pro čtení a zápis dat existují speciální funkce. Pokud uživatel chce číst z databáze, musí nastavit tzv. *posluchače* (listener) na určitou část z databáze. Posluchač poté uživateli předá tuto část i se všemi zanořenými daty. Předání je spuštěno určitou událostí, např. změnou části databáze nebo modifikováním synovského elementu. Poslední možností je vyvolání dat bez čekání na událost. V takovém případě však nejsou informace dostupné okamžitě, ale s menším zpožděním. Dokumentace [3] proto doporučuje jakoukoliv práci se získanými daty provádět uvnitř funkce pro čtení. Program totiž nečeká na to, až data dorazí, ale přejde ihned na další řádek kódu. Může se pak stát, že program bude chtít pracovat s daty, která však ještě nejsou k dispozici.

Databáze taktéž umožňuje práci v offline režimu, když není dostupné internetové připojení. Firebase pak pracuje pouze s daty, která jsou uložena v *pomocné paměti* (cache). V případě provádění dotazů nad databází jsou tyto příkazy uloženy a po obnovení internetového připojení provedeny ve stejném pořadí, v jakém byly zadány.

## Autentizace

Možnost přihlášení a registrace uživatelů je velice snadná, protože je zapouzdřena do několika knihovnických funkcí. Ty přijímají jako parametry uživatelův email a heslo. Voláním

<sup>10</sup>JSON — JavaScript Object Notation

funkce pro registraci s údaji, které už v databázi existují, lze ověřit, zda daný uživatel už je registrován. Kromě přihlašování a registrace přes email jsou rovněž implementovány funkce pro přístup pomocí účtu na Facebooku, Twitteru, GitHubu nebo Googlu.

## Posílání a přijímání zpráv přes cloud

Firestore umožňuje přes cloud posílat zprávy na konkrétní zařízení. Samotné posílání je možné několika způsoby. Základní dělení spočívá v tom, zda má být zpráva poslána jednotlivci nebo skupině. Pro poslání zprávy jednotlivci je třeba jako klíč **"to"** uvést speciální *token* příjemce, který se získá funkcí `FirebaseInstanceId.getInstance().getToken()`. Posílání zpráv na více zařízení se dělí na dva způsoby. Zařízení může být registrováno k odběru určitého *tématu* (topic) a tedy každá zpráva s tímto označením bude doručena všem odběratelům. Každé zařízení pak může být registrováno pro více témat. Druhý způsob je posílání zpráv určité skupině zařízení. Uživatel ve zprávě definuje skupinu zařízení a pošle tuto zprávu serveru. Ten jako odpověď pošle *notifikační klíč* dané skupiny. Zařízení se do této skupiny přihlásí právě prostřednictvím daného klíče a unikátního identifikátoru, který si uživatel zvolí. Rozdíly mezi strukturami zpráv<sup>11</sup> pro jedno zařízení resp. pro odběr tématu jsou ukázány na listingu 4.1 resp. 4.2.

```
https://fcm.googleapis.com/fcm/send
Content-Type: application/json
Authorization: key=AIzaSyZ-1u...0GBYzPu7Udno5aA

{ "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
  "data" : {
    "message": "Zprava pro konkretniho uzivatele."
  }
}
```

Listing 4.1: Příklad struktury zprávy pro poslání konkrétnímu uživateli. Na prvním řádku je adresa serveru, druhý udává formát zprávy a na třetím je autorizační klíč, který je unikátní pro každou aplikaci ve Firestore a odlišuje tak zprávy jiných aplikací. Klíč **"to"** obsahuje token příjemce.

```
https://fcm.googleapis.com/fcm/send
Content-Type: application/json
Authorization: key=AIzaSyZ-1u...0GBYzPu7Udno5aA

{ "to": "/topics/nas_projekt",
  "data": {
    "message": "Posilam tento dulezity text.",
  }
}
```

Listing 4.2: Příklad struktury zprávy pro poslání na konkrétní téma, kde klíč **"to"** obsahuje jméno tématu, kterého se zpráva týká.

V reakci na odeslanou zprávu server pošle odpověď, zda byla zpráva úspěšně doručena. Odpověď se liší podle způsobu odeslání. Pokud byla zpráva odeslána pro určité téma, v případě úspěchu odpověď obsahuje identifikátor dané zprávy. Jestliže odeslání nebylo úspěšné,

<sup>11</sup>Uvedené struktury JSON formátu jsou upraveny z <https://firebase.google.com/docs/cloud-messaging/send-message>.

zpráva bude obsahovat chybovou hlášku, informující o důvodu nezdaru. Mezi důvody může patřit špatně zadaný token při posílání na konkrétní zařízení, nesprávně definovaný formát JSON struktury zprávy nebo špatně zadaný parametr zprávy. Pro posílání zprávy skupině zařízení následuje odpověď, která udává, na kolika z nich byla zpráva úspěšně přijata a na kolika nikoliv.

Pro přijímání zpráv je nutné přepsat metodu `OnMessageReceived()`, v jejímž parametru `remoteMessage` je obsažena přijatá zpráva.

## Úložiště

Slouží pro uchování různých souborů jako je video, obrázky nebo zvukový záznam. Je možné ve Firebase konzoli specifikovat práva pro přístup do úložiště. Soubory jsou ukládány do *Google Cloud Storage* (GCS), čímž jsou dostupné jak přes Firebase, tak i přes GCS. Pokud je dostupnost internetového připojení omezena, je možné stahování dokončit přímo od místa přerušení. Úložiště může být rozděleno do několika menších instancí, které mohou mít vlastní specifika. Například je možné podle polohy uživatelů určit, aby byla instance blízko těmto uživatelům, čímž se sníží latence. Dále je možné každé instanci určit jiná omezení např. podle toho, jaké typy souborů obsahují. Získávání souborů probíhá podobně jako v případě databáze, kdy přístup k souboru je pomocí odkazu na něj. Při ukládání souboru lze přidat metadata jako je velikost souboru, jméno nebo jeho typ. Proces uložení lze řídit pomocí funkcí `pause()`, `resume()` nebo `cancel()`, které jej přeruší, nechají pokračovat nebo ukončí.

Soubory lze stáhnout několika způsoby. Může se získat např. jako pole bytů, jako soubor nebo pomocí *FirebaseUI*<sup>12</sup> a knihovny *Glide*<sup>13</sup>. Stahování probíhá na pozadí aktivity, i když se stav aktivity mění.

Pro úložiště lze také specifikovat bezpečnostní pravidla, která určují, kdo smí ukládat a stahovat soubory. Příklad nastavení pravidel tak, aby k souborům mohli přistupovat jen přihlášení uživatelé, je ukázán na listingu 4.3. Pomocí `match` se specifikuje cesta k souboru, pro který má pravidlo platit a v proměnné `request` jsou uloženy informace o uživateli, který se souborem manipuluje, a o souborech, kterých se manipulace týká. Díky tomu lze např. ověřit, zda soubor k uložení je požadovaného typu nebo nepřekračuje uživatelem stanovenou maximální velikost. Pro umožnění přístupu všem uživatelům stačí odstranit podmínku v `allow`. Tato sekce, včetně ukázky práv, byla převzata z Firebase Storage dokumentace [5].

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Listing 4.3: Bezpečnostní pravidla, která dovolují pro všechny složky a soubory zápis a čtení jen přihlášenému uživateli, což je zaručeno, pokud `request.auth` není `null`.

<sup>12</sup>FirebaseUI — jedná se o knihovnu, která slouží k propojení grafických elementů s Firebase

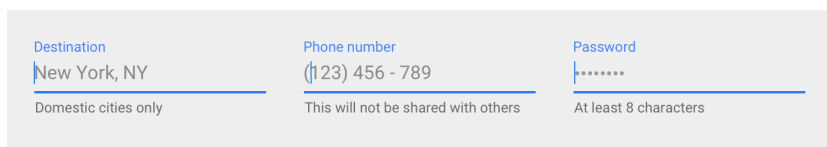
<sup>13</sup>Glide — <http://bumptech.github.io/glide/>

## 4.3 Material Design



Obrázek 4.3: Příklad Material designu pro přesné zvolení koordinace řádku s textem a nadpisem.

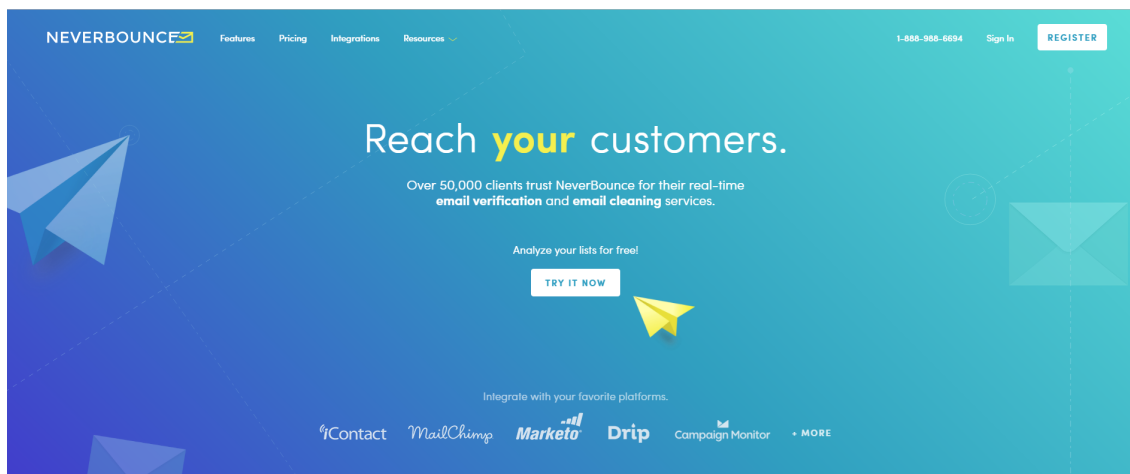
Jedná se o soupis poznatků a návrhů pro vytvoření jednotného a přehledného grafického rozhraní mobilní aplikace. Postupně se začlenil do všech zařízení, které mají operační systém Android (tablety, Chromebooky). Dalo by se říci, že se jedná o uživatelskou příručku [7], která popisuje, jak správně tvořit design jednotlivých komponent, např. menu, layout, textový box, tak i doporučení, kdy je vhodné použít kterou komponentu. Na obrázku 4.3 je ukázáno, jaké jsou například doporučované rozestupy mezi textovým polem a jeho nadpisem. Jak je vidět, je mezi nimi rozestup 8dp<sup>14</sup>. Na obrázku 4.4 je zase příklad uzpůsobení popisků pro co nejlepší informovanost uživatele.



Obrázek 4.4: Náhled na řádky textu, které se skládají ze dvou popisků. Horní nadpis udává, co má uživatel vyplnit a dolní popisek sděluje další informace, které by měl uživatel vědět.

<sup>14</sup>dp — density-independent pixels. Doporučuje se tato jednotka používat pro rovnoměrné zobrazení elementů při různých hustotách obrazovek. Obrázky jsou převzaty z <https://material.io/guidelines/components/text-fields.html#text-fields-layout>.

## 4.4 Poznatky o tvorbě designu aplikací



Obrázek 4.5: Příklad webové stránky, která má pěkný a přehledný design. Uprostřed je stručně popsáno, o čem tato stránka je a co má udělat. Menu je situováno na kraj, aby neodvádělo pozornost od hlavní funkce stránky.

Kromě toho, že aplikace musí splňovat veškeré funkce, které si vývojář vysnil, musí se také líbit zákazníkům. Existuje spousta aplikací, které jsou stejně dobré nebo i lepší, ale uživatel si zvolí tu, která má lepší design, je přehlednější a obecně snadnější na ovládání. Tyto principy je třeba ctít jak při vývoji webových, tak i mobilních aplikací. Při vytváření vzhledu aplikace je třeba brát v potaz i psychologii a myšlení uživatele. Důležitý je první kontakt s aplikací, který z větší části rozhodne o tom, zda si uživatel zvolí právě tento program. Steve Krug ve své knize [11] upozorňuje na to, jak důležité je udělat aplikaci přehlednou a co nejjednodušší. Uživatelé nechtějí přemýšlet a při jakékoliv práci navíc ztrácejí o aplikaci zájem. Důležité je také brát v potaz rozmístění prvků na obrazovce. Podle článku o vlivu psychologie na design [13] uživatelé lépe reagují na aplikace, které jsou navrženy podle určitého vzoru/rozestavení, se kterým se již v minulosti setkali a tedy se v aplikaci lépe orientují. Když už uživatel aplikaci najde, v několika málo okamžicích se rozhoduje o tom, zda pro něj má přínos, nebo jestli se k ní ještě někdy vrátí. Jak docílit toho, aby se zákazník rozhodl aplikaci využít a aby tato dostatečně upoutala pozornost, popisuje Alexander Dawson ve své knize [8]. Také udává, že je dobré se inspirovat aplikacemi, které svým osobitým designem získaly velkou oblibu mezi lidmi, čehož jsem využil, jak již bylo zmíněno v kapitole 3.

Za základě těchto poznatků jsem navrhl mobilní a webovou aplikaci, jak je popsáno dále v kapitole 5. První stránka by byla čistě pro sdělení základních informací o aplikaci s odkazem na její stažení na Google Play<sup>15</sup>. Druhá stránka bude plnit funkci nástěnky s grafy jednotlivých projektů zobrazených vedle sebe. Na aplikaci tedy nebude nic složitého, nad čím by musel uživatel přemýšlet. Na obrázku 4.5 je příklad stránky, která slouží čistě pro informování návštěvníků a nenabízí žádnou složitou funkcionalitu, na kterou by uživatel nemusel přijít<sup>16</sup>. Právě tuto jednoduchost a schopnost upoutat pozornost uživatele jsem se snažil předat svojí aplikaci.

<sup>15</sup> Google Play — služba společnosti Google, přes kterou si uživatelé mohou stáhnout aplikace pro Android

<sup>16</sup> Obrázek stránky byl převzat od společnosti NeverBounce — <https://neverbounce.com/>.



## 4.5 Interakce webové a mobilní aplikace

Pro snadnou spolupráci mezi webovou a mobilní aplikací jsem zvolil platformu Firebase. Vyhovuje totiž přesně požadavkům, které pro svoji aplikaci vyžaduji. Její databáze umožňuje aktualizaci dat při jakékoliv změně, podporuje vývoj jak pro mobilní zařízení, tak i pro webové aplikace a nabízí spoustu užitečných funkcí, jako je ukládání dat do cloudu, posílání zpráv přes cloud a nástroje pro testování. Jelikož je komunikace s databází kromě *JavaScriptu* a *Node.js* také možná pomocí *REST API*<sup>17</sup>, které komunikuje pomocí *HTTP*<sup>18</sup> příkazů a data mezi klientem a serverem jsou vyměňována přes JSON struktury, mohou být server i klient aplikace implementovány různými jazyky, které umožňují zpracování JSON struktur. Moje webová aplikace však slouží pouze pro zobrazení dat z databáze, a proto jsem nemusel implementovat serverovou část.

Jako jedna z alternativ se nabízí *Amazon AWS*<sup>19</sup>, který disponuje podobnými funkcemi. Firebase se mi však zdálo vhodnější pro malou a jednoduchou aplikaci. Dále se nabízí *Realm*<sup>20</sup>, který oproti Firebase umožňuje mít lokální databázi v mobilu. Tu však nakonec nebylo nutné vytvářet, jelikož Firebase umožňuje ukládat data do vyrovnávací paměti, ze které data načítá v případě výpadku síťového připojení.

## 4.6 Google Play konzole

Nahrání aplikace na Google Play je možné pomocí její konzole<sup>21</sup>, která je podobná té od Firebase. Kromě zveřejnění aplikace shromažďuje i statistiky o jejím používání a poskytuje mnoho podpůrných funkcí. Uživatel může například sledovat nárůst nebo pokles aktivních uživatelů nebo počet instalací, odinstalací a aktualizací. Užitečnou funkcí je i hlášení o chybách na uživatelských telefonech např. kdy aplikace spadla, na jaké obrazovce k tomu došlo, která funkce pád způsobila nebo jaký telefon s jakou verzí Androidu měl dotčený uživatel. Veškeré statistiky jsou vykresleny do grafů. Je možné svoji aplikaci propojit s některými dalšími funkcemi, jako je Firebase Cloud Messaging, díky čemuž mohou být informace o posílání zpráv zahrnuty do statistik.

Dále jsou pro vývojáře uchovávány dřívější verze APK, které si může uživatel stáhnout nebo si může projít telefony, které aplikace podporuje. Důležité informace pro vývojáře potom jsou reporty z automatického testování aplikace na deseti zařízeních s různými verzemi Androidu od 5.0 až po 8.0. Výsledkem testů jsou poté statistiky o vytížení procesoru, o přenosu dat, snímky obrazovky a video, jak přesně bylo postupováno při testování.

Je také možné upravit věci, které vývojář nastavoval při prvním publikování aplikace, jako jsou grafické obrázky nebo popis funkcionality. Další z možností je zobrazení recenzí a hodnocení aplikace ostatními uživateli. Nakonec je možné nastavit, jak má být aplikace propagována. Ukázka konzole je na obrázku 4.6.

<sup>17</sup>REST — Representational State Transfer

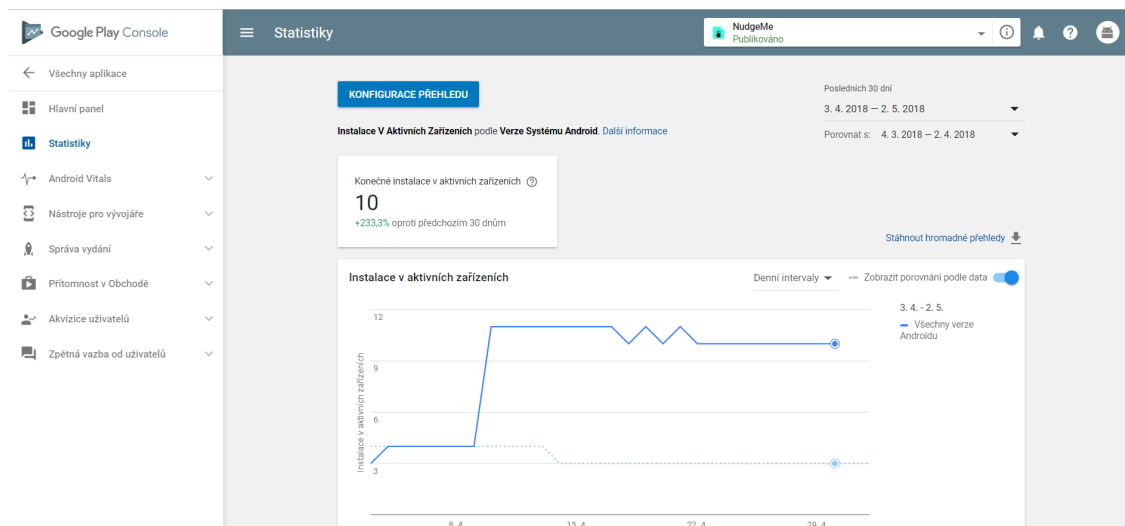
<sup>18</sup>HTTP — Hypertext Transfer Protocol

<sup>19</sup>Amazon Web Services — <https://aws.amazon.com/mobile>

<sup>20</sup>Realm — <https://realm.io>

<sup>21</sup>Google Play konzole — <https://play.google.com/apps/publish/>





Obrázek 4.6: Google konzole ve které je na levé straně menu sestávající se z různých možností pro vývojáře, nahoře je zobrazena aktuálně používaná aplikace a uprostřed je graf zobrazující, jak se v čase měnil počet instalací na aktivních zařízeních. Pro tento graf lze nakonfigurovat i jiné statistiky, které je také možné si stáhnout.

## 4.7 Zveřejnění aplikace na Google Play

Nejjednodušší způsob jak aplikaci dostat k uživatelům, je zveřejnit ji na Google Play. K tomu je třeba mít účet na Googlu a zaplatit jednorázový poplatek \$ 25<sup>22</sup>. Po jeho zaplacení se přechází k vyplnění informací o aplikaci. Je nutné vyplnit dotazník o aplikaci, ve kterém jsou otázky typu, zda aplikace je vhodná pro děti nebo jestli neobsahuje nevhodné obrázky. Dále je nutné přečíst a podepsat dokument *Zásady pro vývojáře* [6], kde jsou další informace o tom, jaká pravidla by měla aplikace dodržovat. Následně je třeba nastavit grafické prvky, které se budou zobrazovat v Google Play u aplikace. Jedná se především o název aplikace, obrázky jednotlivých obrazovek a krátký popis funkčnosti a účelu. Je možné zvolit distribuční země a vymezit typy zařízení, na které se může aplikace instalovat. Potom je třeba vybrat, do jaké kategorie aplikace patří (produktivita, hra, nakupování) a minimální věkovou hranici uživatele. Nakonec je třeba rozhodnout, zda aplikace bude placená či zdarma. Pokud se zvolí možnost bez poplatku, v budoucnu již nebude možné ustanovit aplikaci jako placenou.

Je možné před ostrým vydáním zvolit možnost alfa nebo beta testování. V rámci nich je možné se rozhodnout mezi otevřeným a uzavřeným testováním.

- uzavřené testování — vydavatel vytvoří seznam testerů a zašle jim odkaz, přes který se dostanou na stránku Google Play, kde potvrdí, že se chtějí stát testerem a budou moci si aplikaci stáhnout
- otevřené testování — vydavatel nemusí vytvářet seznam testerů, ale stačí poslat odkaz na aplikaci, díky tomu může aplikaci testovat více lidí pouhým přeposláním odkazu dalším lidem

<sup>22</sup>Tento údaj je platný ke dni 14.5.2018.

- alfa verze — zahrnuje úzkou skupinu testerů, od března 2018 již není možné využívat otevřené alfa testování
- beta verze — zpravidla následuje po alfa testování, širší spektrum testerů, je možné zvolit otevřené testování

Pro nahrání aplikace je nutné vložit do konzole podepsaný APK, který lze vytvořit přes Android Studio. Pro podepsání je nutné mít klíč ve formátu *JKS*<sup>23</sup>. APK je nutné podepsat některým z certifikačních schémat *V1* nebo *V2*. *V1* je především pro starší zařízení, která nepodporují *V2*. Slouží pro podepsání *JAR*<sup>24</sup> souborů, ale nedokáže ochránit některé další soubory, které jsou součástí APK, jako jsou např. metadata v .zip souborech. Naopak *V2* tyto problémy odstraňuje, a proto se doporučuje APK podepsat oběma certifikáty. Při generování nové verze APK, je nutné ji podepsat vždy stejným klíčem. Tento postup jsem převzal z příručky pro Android [2]. Podepsanou verzi potom lze nahrát do konzole vložení do určitého typu vydání (alfa, beta, produkční). Při vytvoření nové verze APK je nutné v souboru `build.gradle` zvýšit kód a jméno verze, jinak dojde k chybě při nahrání aktualizace do konzole. Zvýšení verze je nutné z toho důvodu, aby nebylo možné nahrát jako aktuální verzi starší verzi aplikace. Jméno verze slouží čistě k informování uživatele. Po potvrzení vydání je aplikace testována na několika zařízeních, a proto vydání chvíli trvá.

---

<sup>23</sup>JKS — Java Key Store je formát souboru pro uložení kryptografických klíčů nebo certifikátů

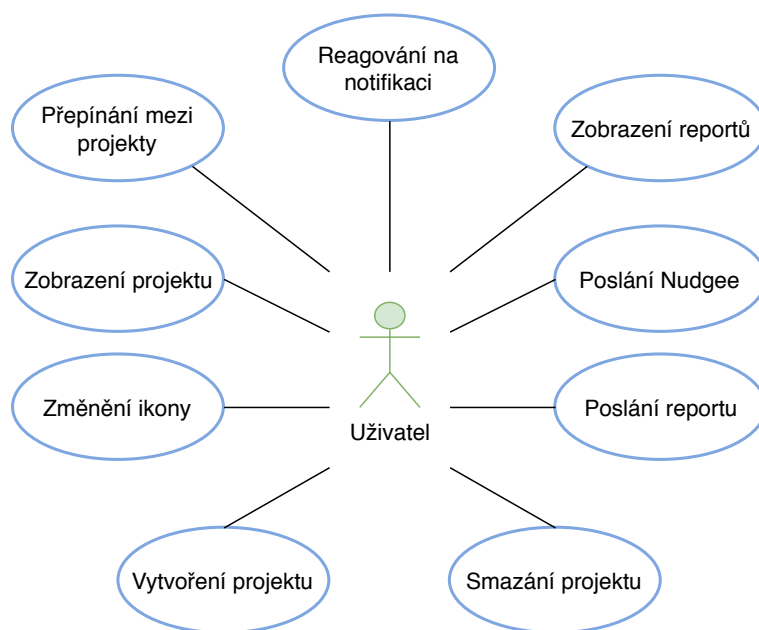
<sup>24</sup>JAR — Java ARchive

## Kapitola 5

# Návrh aplikace NudgeMe

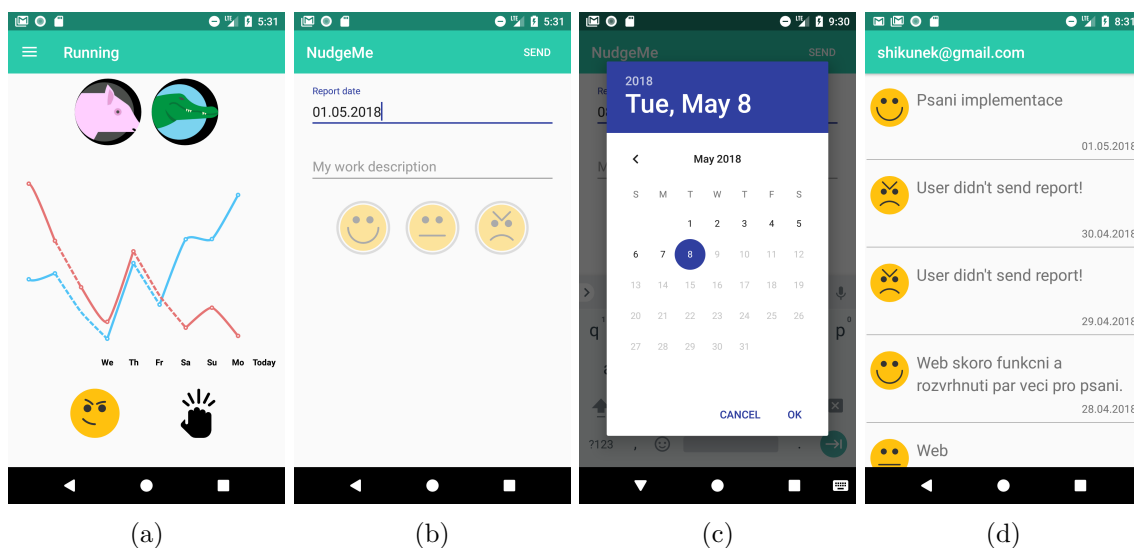
Kapitola se věnuje důkladnému popsání funkcionality, návrhu a interakce aplikace s uživatelem. Nejprve je vše popsáno pro mobilní aplikaci včetně naimplementované funkcionality a historie verzí. Dále následuje popis struktury databáze, a proč jsem ji takto navrhl. Nakonec je popsána webová aplikace včetně návrhu vzhledu.

### 5.1 Popis rozložení obrazovek mobilní aplikace



Obrázek 5.1: Use case diagram popisující hlavní funkce aplikace, které může uživatel použít. Kromě nich je také umožněno reporty přepisovat, odebírat členy z projektu, upravovat projekty nebo si nastavit uživatelské jméno.

Jak již bylo zmíněno v sekci 2.1, mezi základní funkce aplikace patří zobrazení grafu a posílání reportu a Nudgee. Pro lepší přehled jsem se rozhodl tyto a některé další funkce vykreslit pomocí Diagramu případů použití (Use Case diagramu), který patří mezi nástroje pro vývoj, a který je možné vidět na obrázku 5.1. Jeho detailnější charakteristika a popis výhod jsou popsány v knize [9] od Martina Fowlera a Kendalla Scotta.

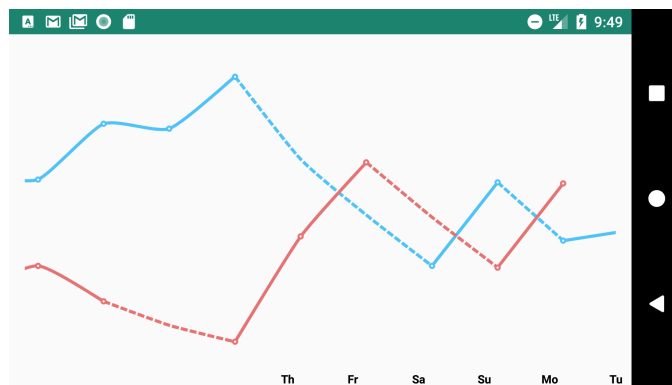


Obrázek 5.2: Náhled konečné podoby aplikace. Po kliknutí na tlačítko se smajlíkem na obrazovce 5.2a se dá dostat na obrazovku 5.2b. Zde kliknutím na datum se objeví kalendář, který je zobrazen na obrázku 5.2c. Na něm je možné vidět, že budoucí dny jsou zašedlé a tedy není možné je zvolit. Na posledním obrázku je výpis reportů konkrétního uživatele.

Když se uživatel úspěšně přihlásí, ukáže se mu hlavní obrazovka 5.2a, zobrazující uživatelův aktuálně vybraný projekt. Kromě grafu ukazujícího intenzitu práce jednotlivých členů, zobrazuje i ikony všech spolupracovníků, kteří se na daném projektu podílejí. Graf je v jistých ohledech interaktivní a umožňuje přiblížení, resp. oddálení nebo posun doleva, čímž se zobrazí hodnoty reportů z dřívějšího období. Pokud uživatel otočí hlavní obrazovku naležato, zobrazí se pouze graf, který pokryje celou plochu displaye a umožní tak větší přehlednost, jak je ukázáno na obrázku 5.3. Domnívám se, že tento princip je vhodný pouze pro tuto obrazovku a na ostatních by neměl takový efekt. Pomocí uživatelových obrázků, které jsou, jak již bylo zmíněno, inspirovány aplikací Messenger, je možné se dostat na výpis reportů konkrétního uživatele, který je zobrazen na obrázku 5.2d.

Tlačítko NudgeMe na hlavní obrazovce 5.2a slouží pro posílání zprávy všem uživatelům, kteří v aktuální den nezadali svůj report. V reakci na zprávu se uživateli, který report nezadal, zobrazí notifikace, která umožní, po kliknutí na ni, přejít ihned na obrazovku 5.2b pro vyplnění reportu. Upozornění přijde na každé zařízení, na kterém je uživatel aktuálně přihlášen. V případě, že již má určitý uživatel vyplněn report na tento den a někdo jiný pošle zprávu, nebude uživateli, který report vyplnil, zobrazena tato notifikace. Jedná se o podobnou funkci, jako je "šťouchnutí" přátel na Facebooku, díky kterému se uživateli zobrazí notifikace. Přejít na zadání reportu je možné také pomocí tlačítka se smajlíkem na hlavní obrazovce.

Obrazovka 5.2b zobrazuje, jaké informace je třeba vyplnit v reportu. Nejprve je to datum, které je primárně nastaveno na aktuální den, dále textový blok pro napsání krátké zprávy informující ostatní o provedené práci a výběr *smajlíka* symbolizujícího pocit z odvedené práce. Pokud se stane, že uživatel v předchozích dnech nevyplnil reporty, po odeslání reportu na aktuální den se automaticky dogenerují reporty do databáze pro nevyplněné dny a budou obsahovat speciální hodnoty, které je odliší od zadaných reportů. Datum reportu lze nastavit kliknutím na řádek s datem, díky čemuž se zobrazí kalendář, který je zobrazen na obrázku 5.2c. V něm jsou omezeny dny, pro které je možné report vyplnit. Nedá se

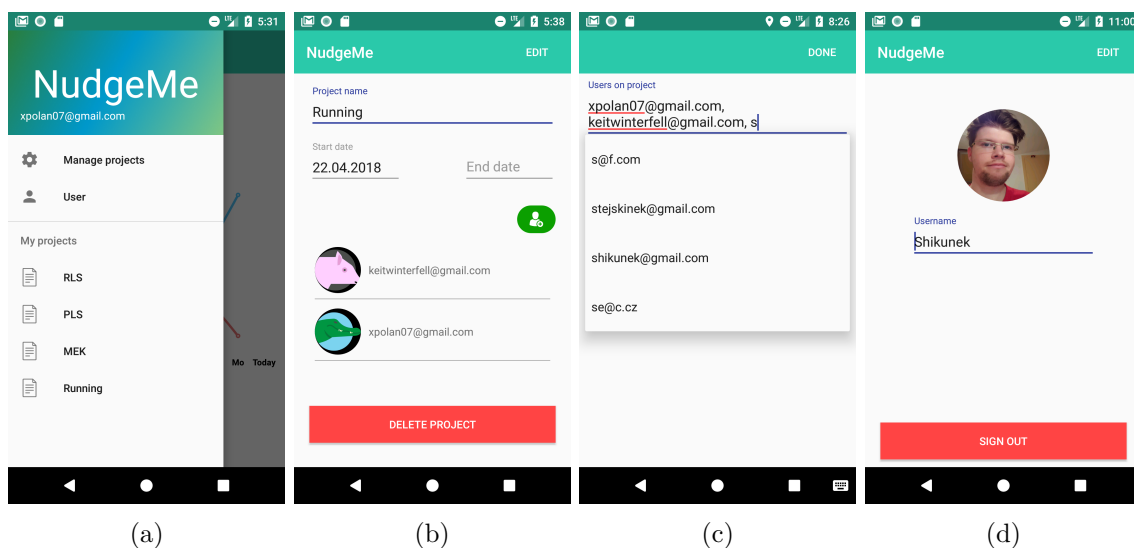


Obrázek 5.3: Náhled hlavní obrazovky při otočení mobilu naležato. Graf je zobrazen přes celou obrazovku a získává tak na přehlednosti.

zvolit den, který předchází dni, kdy byl projekt vytvořen. Toto slouží pro zabránění určitému chaosu v databázi. Řekl bych, že možnost zadat report na dřívější datum je zbytečná a v praxi zřídka kdy použitelná. Dále není možné zadat reporty pro budoucí dny, aby se zajistila větší serióznost reportů. Nikdo totiž neví, co přesně a jak dobře udělá v následujících dnech. Pokud uživatel nevyplní report nebo nezvolí smajlíka, report nebude možné odeslat. Toto omezení sice snižuje rychlost jakou uživatel report vyplní, ale je to v zájmu toho, aby uživatel lépe informoval své kolegy.

Kliknutím na uživatelova avatara se zobrazí jeho seznam reportů 5.2d od nejnovějšího po nejstarší. Každá položka seznamu obsahuje uživatelský popis práce, typ smajlíku, kterým se ohodnotil, a datum zadání reportu. Pokud uživatel nezadal report pro některý den, při dogenerování tohoto reportu je uživateli nastaven smutný smajlík. Učinil jsem tak proto, abych donutil uživatele reporty pravidelně vyplňovat, neboť nezadání reportu je posuzováno stejně, jakoby uživatel nebyl se svojí prací spokojen. Tyto reporty jsou od ostatních odlišeny pomocí přednastaveného textu a navíc jsou vyjádřeny přerušovanou čarou v grafu, jak je možné vidět na obrázku 5.2a.

Pomocí menu, které je zobrazeno na obrázku 5.4a, uživatel může přepínat mezi svými projekty a také se dostat na seznam svých projektů 5.5c. V něm po kliknutí na konkrétní projekt se zobrazí jeho podrobnější informace zobrazené na obrázku 5.4b. Jediný parametr, který nelze upravit ani smazat, je datum vytvoření projektu. Domnívám se, že umožněním změny data vytvoření by se princip aplikace zbytečně zkomplikoval a toto datum tedy slouží čistě pro informování uživatelů. Zadáním data konce projektu nedojde k žádné akci; toho datum je pouze připraveno pro případné budoucí použití. Uživatel také může projekt smazat. Nedojde však ke smazání celého projektu a všech uživatelů, ale pouze ke smazání hodnot uživatele, který takto učinil. Projekt tedy existuje v databázi tak dlouho, dokud si jej neodstraní všichni uživatelé, kteří se na něm podílí. Po kliknutí na zelené tlačítko se dá dostat na obrazovku 5.4c. Uživatel zde pomocí našeptávače může vybrat uživatele podle jeho emailu a tím ho přidat k projektu. Do našeptávače jsou načteny emaily všech uživatelů, kteří se v aplikaci registrovali. Pokud bude zadán email, který nepatří žádnému registrovanému uživateli, objeví se chybová hláška informující, který email je nevalidní a nebude možné tento seznam uživatelů potvrdit tlačítkem Add.

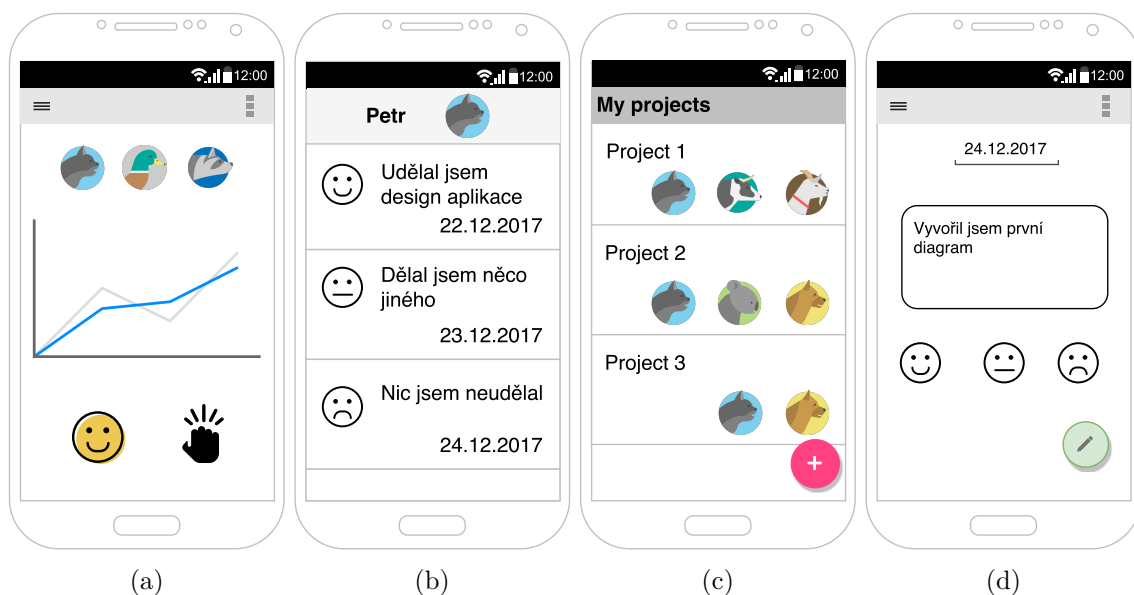


Obrázek 5.4: Seskupení obrázků, které zobrazuje zbývající obrazovky finálního vzhledu. Na první obrázku je možné vidět menu, které se sestává z názvu aplikace, emailu přihlášeného uživatele, jeho seznamu projektů a možnosti přejít na svůj seznam projektů nebo na svoje nastavení. Dále je obrazovka zobrazující informace o konkrétním projektu, přes kterou se dá dostat na přidávání a odebírání uživatelů na projekt 5.4c. Poslední obrázek 5.4d zobrazuje nastavení uživatele s uživatelským obrázkem, kdy po kliknutí na obrázek může nahrát nový ze svého telefonu, a s možností zadat svoje uživatelské jméno.

## Popis rozdílů mezi jednotlivými verzemi aplikace

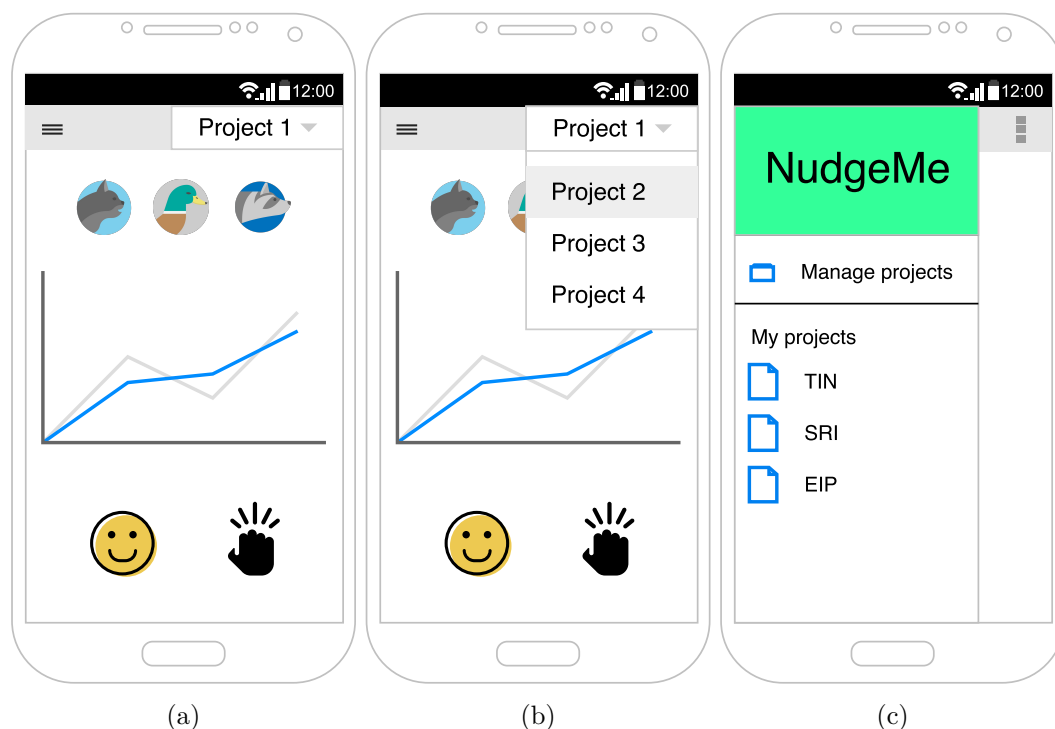
Jednotlivé verze programu se hlavně liší svoji grafickou podobou, protože již v první verzi byla implementována takřka veškerá základní funkcionalita, jako je posílání reportu, vyznačení reportů do grafu a posílání Nudgee. Na začátku jsem vytvořil základní rozložení obrazovek pomocí Mockup modelu, který usnadňuje návrh designu programu, jak je uvedeno v článku [14]. Můj Mockup model hlavních obrazovek je možné vidět na obrázku 5.5. Rozložení obrazovek a jednotlivých prvků zůstalo do jisté míry zachováno až do finální verze. Upustil jsem od uživatelské ikony v seznamu reportů 5.5d, protože mi přišlo, že by obrazovka s obrázky smajlíků u reportů byla až příliš barevná. Historie přepínání mezi projekty (viz. obrázek 5.6), se postupem času zdokonalovala až do současného návrhu obrazovky 5.6c, která, jak již bylo zmíněno, byla inspirována aplikací Slack. Mezi druhou a finální verzí došlo k obměně designu, kdy původní návrh 5.7 byl nahrazen Material Designem. Došlo ke zvětšení ikonky uživatelů a tlačítka pro report a posílání Nudgee jsou rozestavena více od sebe a přichycena ke spodní části obrazovky.

Dále jsem zavedl obrazovku 5.4d pro nastavení uživatele, kde si každý uživatel může zvolit svoji vlastní ikonu a nastavit svoje uživatelské jméno. To zatím nebylo využito, protože není zaručeno, že každý uživatel si svoji přezdívku vyplní. Mohlo by se povinně vyplňovat při registraci, ale tím by zase registrace trvala o něco déle. Proto je v aplikaci pro sdělení, o jakého uživatele se jedná, zvolena emailová adresa, která je pro každého unikátní. V původním návrhu na obrázku 5.7a měl každý uživatel přidělen určitý avatar, který měl stejnou barvu jako jejich příslušná křivka v grafu. V současné podobě je možné avatara nahradit vlastním obrázkem pro lepší určení, o jakého uživatele se jedná. Pro zdokonalení orientace bude ještě potřeba ikony ohraničit kulatým rámečkem, jehož barva by symboli-

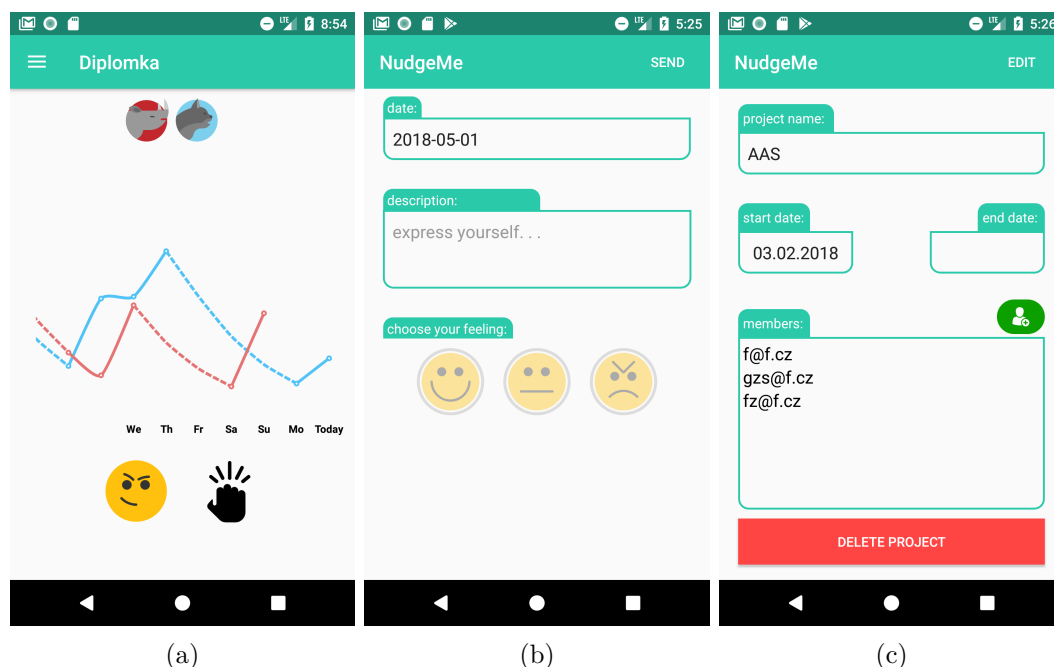


Obrázek 5.5: Návrh hlavních obrazovek aplikace. Na hlavní obrazovce 5.5a je vyobrazen graf, obrázky avatarů všech uživatelů na projektu a tlačítka pro přechod na obrazovku 5.5d a pro posílání Nudge. Na dalším snímku 5.5b je výpis reportů uživatele, kde je dále uvedeno jeho jméno a příslušný avatar. Snímek 5.5c zobrazuje projekty, kde u každého z nich se nachází jeho název a avataři příslušných členů. Navíc je k dispozici tlačítko pro přidání nového projektu. Nakonec je snímek 5.5d pro vyplnění reportu.

zovala, ke které křivce patří. Ikony uživatelů byly také implementovány do obrazovky 5.4b, aby byl přehled výraznější.



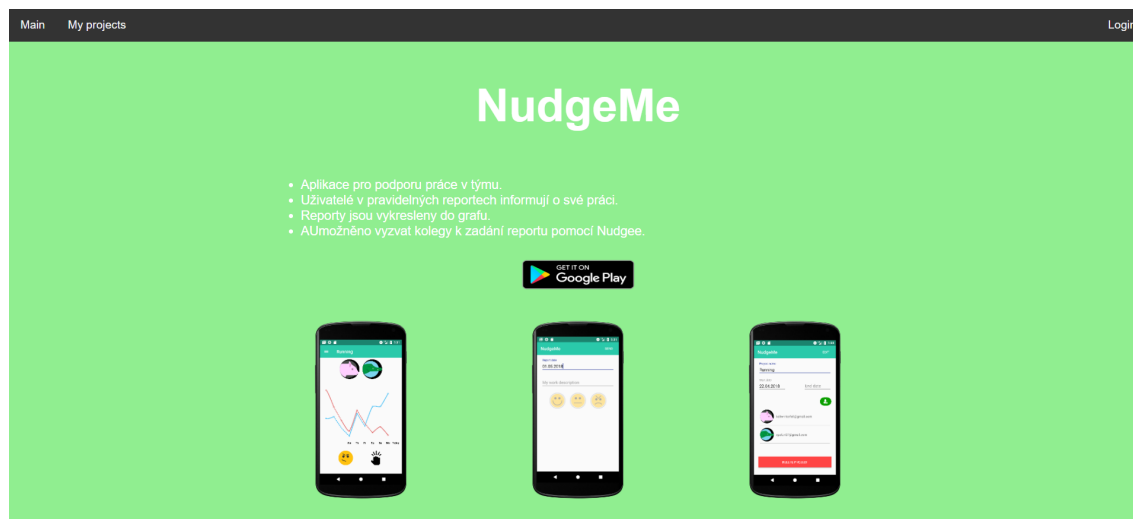
Obrázek 5.6: Zobrazení vývoje principu přepínání mezi projekty. Obrázky 5.6a a 5.6b ukazují použití pomocí rolovacího menu. Nakonec bylo zvoleno přepínání z obrázku 5.6c pomocí bočního menu, které zobrazuje všechny projekty a po kliknutí na jeden z nich se hlavní obrazovka aktualizuje. Tento návrh vyčistil obrazovku. Přepínání projektů probíhá stejně rychle jako předtím, pomocí pouhých dvou kliknutí.



Obrázek 5.7: Kromě obrázků 5.7a a 5.7b, které jsou udělány podle návrhu 5.5 je zde obrázek 5.7c, který znázorňuje detailnější informace o projektu. Pomocí zeleného tlačítka nad seznamem uživatelů se dá dostat na obrazovku pro přidání nebo odebrání uživatelů na projektu.

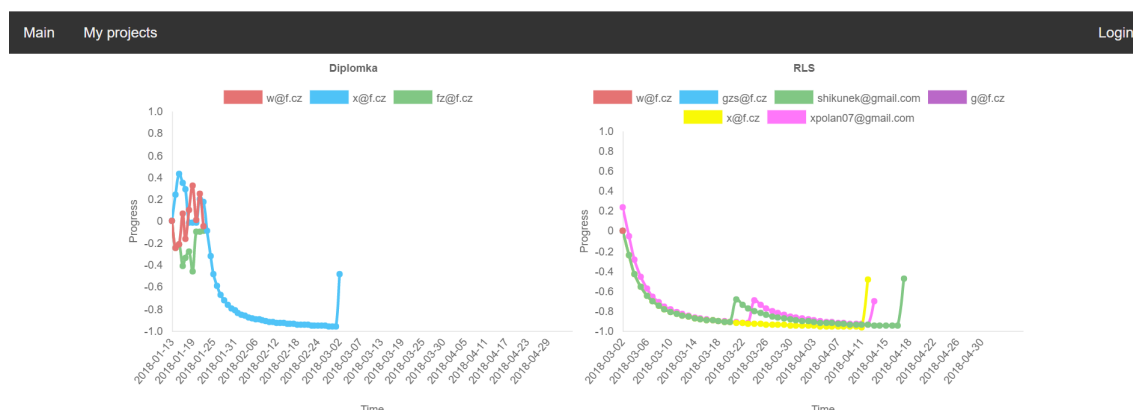


## 5.2 Webová aplikace



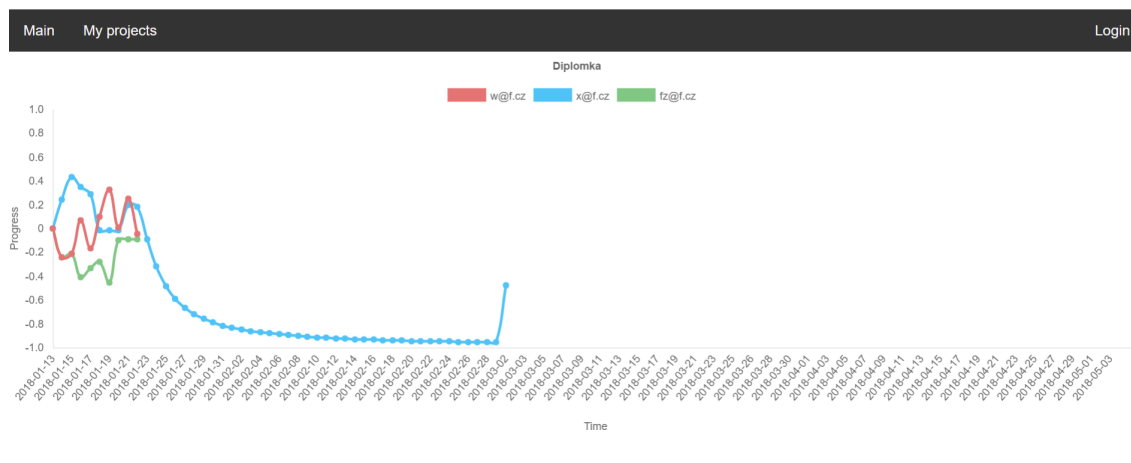
Obrázek 5.8: Návrh úvodní stránky webové aplikace. Sestává se z krátkého výčtu funkcionality, obrázků z mobilní aplikace a odkazu na stažení aplikace na Google Play.

Jelikož primární částí celkového cíle, který má za úkol zpřehlednit a zlepšit přehled o práci v týmu, je mobilní aplikace, webová část má za cíl pouze lépe zobrazit vyplněné reporty. Úvodní část na obr. 5.8 slouží pro krátké sdělení návštěvníkovi, o jakou aplikaci se jedná, s možností přes odkaz na Google Play si ji stáhnout. Tento jednoduchý návrh má za cíl usnadnit uživatelskou orientaci a nezatěžovat je nepodstatnými fakty. Pro uživatele, který aplikaci již používá, je zde možnost se na webu přihlásit, čímž se objeví možnost podívat se na svoje projekty. Oproti mobilní aplikaci není možné se přes web registrovat ani přihlásit pomocí Google účtu. O registraci jsem neuvažoval, protože bych rád, aby uživatelé co nejvíce a primárně používali mobilní verzi. Druhá funkcionality je v budoucím plánu implementace.



Obrázek 5.9: Obrázek webové aplikace se zobrazením všech uživatelských projektů. Každému popisu uživatele s jeho emailem a barvou odpovídá příslušná křivka.

Při přejití na stránku s uživatelskými projekty dojde k jejich automatickému vykreslení jak je ukázáno na obrázku 5.9. Hlavním cílem je umožnit uživateli vidět více svých projektů



Obrázek 5.10: Obrázek webové aplikace, kdy je zvětšen náhled konkrétní aplikace. Graf se roztáhne přes celou šíři obrazovky, čímž dojde ke zpřehlednění dat.

vedle sebe a díky tomu porovnat jejich průběh. Jedná se o prvotní koncept, kdy grafy s příliš mnoha daty jsou zúženy na pevnou velikost. Přehlednější zobrazení je možné po kliknutí na graf, čímž se roztáhne přes celou šíři obrazovky (viz. obrázek 5.10). Dojde tak k přehlednějšímu zobrazení dat reportů a usnadnění orientace v zadaných hodnotách v jakém dni byly zadány. Proto jsem také oproti mobilní aplikaci x-ovou osu popsal daty místo popisu, o jaký den v týdnu se jedná.

### 5.3 Strukturování dat v databázi

Databáze aplikace se dá z relačního pohledu strukturovat do tří entit. Jsou jimi projekt, uživatel a report. Jelikož jsou data ve Firebase databázi uloženy jako JSON struktury, je nutné strukturovat entity podle jiného postupu. Podle doporučení z dokumentace Firebase [3], aby data nebyla zanořena do příliš velké hloubky, jsem databázi rozdělil do 2 hlavních struktur. První z nich je na listingu 5.1 a jedná se o strukturu uživatele. Druhá z nich se týká projektu a je na listingu 5.2. První struktura má u sebe uloženy identifikátory všech uživatelových projektů, přes které se lze dostat k detailnějším informacím. Dobrovolným parametrem je potom uživatelova přezdívka. Datum vytvoření projektu není uvedeno, poněvadž se dá snadno získat, jako datum prvního reportu. Struktura projektu má v sobě uloženy všechny reporty, které jsou identifikovány podle určitého uživatele a data vyplnění.

```

{ "Uzivatel" : {
  "QXKspxCJ22eKnBO9ga4tAOGAI8B2" : {
    "Aktivni" : "-L066PlxBVXxmgTkH0Bz",
    "Projekty" : {
      "-L065iHIcSW8enb5txpZ" : {
        "Nazev" : "TIN"
      },
      "-L066PlxBVXxmgTkH0Bz" : {
        "Nazev" : "SRI"
      }
    },
    "Email" : "gzs@f.cz",
    "Prezdivka" : "Shikunek"
  }
}

```

Listing 5.1: Ukázka návrhu struktury dat z Firebase databáze pro určitého uživatele. V tomto případě má uživatel identifikátor "QXKspxCJ22eKnBO9ga4tAOGAI8B2". Snímek ukazuje, že uživatel pracuje na projektech "TIN" a "SRI", přičemž poslední z nich s identifikátorem "-L066PlxBVXxmgTkH0Bz" má aktuálně zvolen jako aktivní.

```

{ "Projekty" : {
  "-L065iHIcSW8enb5txpZ" : {
    "QXKspxCJ22eKnBO9ga4tAOGAI8B2" : {
      "2017-12-11" : {
        "Report" : "Design aplikace",
        "Hodnota" : 1
      }
    },
    "Nazev" : "ZPO"
  }
}

```

Listing 5.2: Návrh struktury uložení dat pro určitý projekt. "-L065iHIcSW8enb5txpZ" je identifikátor konkrétního projektu a ("QXKspxCJ22eKnBO9ga4tAOGAI8B2") je identifikátor konkrétního uživatele. V tomto případě uživatel do projektu s názvem "ZPO" vyplnil pro den 11.12.2017 report se zprávou "Design aplikace" a ohodnotil se známkou 1, tedy usměvavým smajlíkem.

## 5.4 Návrhy budoucí práce

Pro co největší serióznost a objektivitu reportů bych zavedl možnost jejich ohodnocení ostatními členy týmu. Podobně jak to má Facebook u svých zpráv, ke kterým mohou uživatelé připojovat svoje reakce, tak zde by mohli dát reportu palec nahoru nebo dolů. Toto ohodnocení by se pak zahrnovalo do vykreslování grafu. Nemělo by však dojít k příliš velkému ovlivnění, aby se zadaná hodnota reportu nezměnila na opačnou a tím nevyvstala možnost úmyslně snižovat snahu ostatních. Spíše by to sloužilo pro upozornění uživateli, že jeho ohodnocení není úplně správné.

V aktuální podobě může být zadán pouze jeden report denně. Na druhou stranu člověk, který udělá některý den více práce, by měl možnost takovýto úspěch patřičně zvýraznit.

Hodnoty by se buď přičítaly pro aktuální den, nebo by se zadávaly jako samostatné body do grafu a byly by odlišeny časem zadání. Pravděpodobně by bylo možné zadat jen jeden report za hodinu. Kalendář na obrazovce pro vyplnění reportu by mohl být určitým způsobem zobrazen lépe, aby ještě více usnadnil uživateli vyplnění data a přispěl k lepšímu designu obrazovky. Dále by mohlo být užitečné umožnit poslat report přímo z notifikace, jako je např. možné v Messengeru odpovídat na zprávy.

Aplikace by také mohla umožnit více řídit projekt např. pomocí stanovení určitých časových milníků na projektu, jak je tomu u aplikace Workboard, a jejich vykreslování v grafu. Uživatele, který je momentálně určován pomocí emailu, by mohlo být možné identifikovat pomocí uživatelského jména nebo jeho ikony. Obrázek uživatele by asi byl lepší varianta, poněvadž je nastavován už při registraci, kde je náhodně vybrán z určité skupiny avatarů. Oproti tomu by se přezdívka musela povinně vyplňovat při registraci, čímž by registrace mohla trvat déle a více tak obtěžovat uživatele. Pro první uživatele by se mohl vytvořit krátký návod nebo popis funkcionality aplikace, aby se v ní mohli co nejlépe orientovat a využívat veškeré možnosti, které aplikace nabízí.

## Kapitola 6

# Implementace

Nejprve popisují algoritmus pro vykreslení intenzity práce jednotlivých členů v čase, protože je společný jak pro webovou, tak i mobilní aplikaci. Dále následuje detailnější popis konkrétních funkcí a technik, který je rozdělen na část týkající se mobilní aplikace<sup>1</sup> a na část týkající se webové aplikace<sup>2</sup>. U mobilní zmiňuji, jak pracuji s obrázky uživatelů, jak funguje posílání zpráv a zobrazování notifikací, a popisují práci s databází.

### 6.1 Popis algoritmu pro vykreslení intenzity práce

$$f(s, t) = \begin{cases} \delta(1, s) & \text{if } s > 0 \text{ and } t = 1 \\ \sigma(-1, s) & \text{if } s > 0 \text{ and } t = 0 \\ \zeta(-1, s) & \text{if } s > 0 \text{ and } t = -1 \\ \zeta(1, s) & \text{if } s < 0 \text{ and } t = 1 \\ \sigma(1, s) & \text{if } s < 0 \text{ and } t = 0 \\ \delta(-1, s) & \text{if } s < 0 \text{ and } t = -1 \\ \sigma(1, s) & \text{if } s = 0 \text{ and } t = 1 \\ 0 & \text{if } s = 0 \text{ and } t = 0 \\ \delta(-1, s) & \text{if } s = 0 \text{ and } t = -1 \end{cases} \quad (6.1)$$

$$\delta(p, o) = \frac{\arctan\left(p \cdot \left(p \cdot \left(\tan\left(o \cdot \frac{\pi}{2}\right) + 0, 4\right)\right)\right)}{\frac{\pi}{2}} \quad (6.2)$$

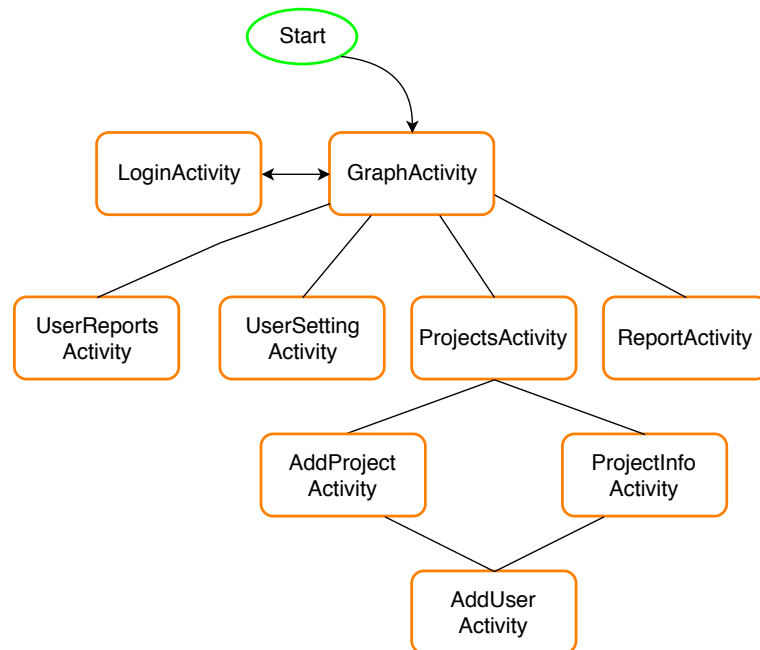
$$\sigma(p, o) = \frac{\arctan\left(p \cdot \left(\tan\left(o \cdot \frac{\pi}{2} + p \cdot \frac{\pi}{2}\right) + 0, 4\right)\right)}{\frac{\pi}{2}} - (1 \cdot p) \quad (6.3)$$

$$\zeta(p, o) = 2 \cdot \left( \frac{\arctan\left(p \cdot \left(p \cdot \left(\tan\left(o \cdot \frac{\pi}{4} + p \cdot \frac{\pi}{4}\right) + 0, 4\right)\right)\right)}{\frac{\pi}{2}} \right) - (1 \cdot p) \quad (6.4)$$

Jak již bylo zmíněno do grafu jsou zaznačeny hodnoty, které si jednotliví uživatelé sami přiřadí při svém ohodnocení v reportu. Hodnoty jsou v rámci aplikace znázorněny jako smajlíci, přičemž spokojený se mapuje na hodnotu 1, neutrální na hodnotu 0 a smutný na hodnotu -1. Nad těmito hodnotami je potom spuštěna funkce 6.1 pro vykreslení grafu,

<sup>1</sup>Mobilní aplikace je dostupná ke stažení na Google Play na adrese <https://play.google.com/store/apps/details?id=com.NudgeMe.petr.testing>

<sup>2</sup>Webová aplikace je dostupná na adrese <http://www.stud.fit.vutbr.cz/~xpolan07/NudgeMe>



Obrázek 6.1: Diagram jednotlivých aktivit znázorňující, že při startu aplikace se nejprve jde do hlavní třídy GraphActivity. Pokud se zjistí, že v aplikaci není přihlášen žádný uživatel, tak dojde k přesměrování na přihlášení/registraci uživatele. Po takto úspěšně provedené operaci se aplikace vrátí zpět na graf. Z ProjectsActivity se lze dostat jak na AddProject tak i na ProjectInfo aktivity. Každá z nich umožňuje přesun na aktivitu pro manipulaci s uživateli.

kde proměnná  $s$  značí předchozí hodnotu na ose  $y$  a proměnná  $t$  udává hodnotu zadaného smajlíka. Funkce 6.2, 6.3 a 6.4 určují, o kolik a jakým směrem se posune další hodnota na ose  $y$ . Parametr  $p$  obsahuje hodnoty 1 resp. -1 podle toho, jestli bude křivka stoupat resp. klesat. Algoritmus má za cíl, aby náhlá změna aktivity uživatele byla výraznější než pozvolná. Pokud uživatel jako předchozí hodnotu zadal -1 a tentokrát 1, bude tato změna vykreslena funkcí 6.4. Dojde tak k prudkému stoupání křivky, což má za cíl ukázat uživateli, že se mu něco povedlo. V opačném případě, tedy když předchozí hodnota byla 1 a následující je -1, dojde k propadu křivky se stejnou intenzitou jako v předchozím případě. Funkce 6.2 a 6.3 fungují na stejném principu, jen se jedná o mírnější změnu. Křivka se díky těmto třem funkcím pohybuje v intervalu  $(-1, 1)$  na  $y$  ose. Zatímco funkce 6.2 a 6.4 směřují nový bod k 1 nebo -1, funkce 6.3 zajišťuje nárůst nebo pokles směrem k 0. Výsledná hodnota vypočítaná některou z těchto tří funkcí je poté přiřazena do proměnné  $s$  a je nad ní spuštěna funkce 6.1 spolu s hodnotou dalšího reportu.

## 6.2 Mobilní aplikace

Každá obrazovka mobilní aplikace je v kódu reprezentována samostatnou třídou. Jejich výčet s vyznačenými závislostmi je vyobrazen na obrázku 6.1. Při spuštění aplikace může dojít k několika situacím. Pokud se jedná o první spuštění a uživatel není přihlášen, uživatel je přesměrován na obrazovku pro přihlášení. Po úspěšném přihlášení nebo registraci mohou nastat dvě situace. Buď uživatel nemá vytvořený žádný projekt a je proto přesměrován

na obrazovku pro vytvoření projektu, nebo alespoň jeden projekt má a je přesměrován na hlavní obrazovku s grafem. Po vytvoření prvního projektu se uživatel ihned dostane na obrazovku s grafem. Je to rozdíl oproti normálnímu vytvoření, kdy je uživatel přesměrován na seznam svých projektů.

Při přihlášení je automaticky uživatel přihlášen k odebírání tématu, které je pojmenované podle uživatelského identifikátoru. Při odhlášení z aplikace je toto předplacení zrušeno. Samotné přihlašování se provádí pomocí Firebase funkcí, přihlášení a registrace pomocí emailu a hesla se provádí funkcemi `signInWithEmailAndPassword(email, heslo)` a `createUserWithEmailAndPassword(email, heslo)`. Přihlášení pomocí existujícího Google účtu se provádí jedinou funkcí `firebaseAuthWithGoogle()` s jedním parametrem, kterým je `googleAccount` s návratovým typem `GoogleSignInAccount`. Je však nutné do Firebase konzole vložit SHA certifikáty z konzole Google Play. Bez nich není toto přihlášení možné. Při přihlášení pomocí Googlu je vždy zobrazena možnost si vybrat, který z účtů chce uživatel zvolit. Původně jsem s touto možností nepočítal, ale v důsledku testování a častým přepínáním mezi účty jsem usoudil, že by tato možnost mohla být užitečná i pro ostatní uživatele.

Při přecházení mezi jednotlivými obrazovkami jsem používal Intenty. Do nich se dají vložit dodatečné informace pomocí funkce `putExtras()`. Ty jsem využil zejména při manipulaci s uživateli na projektu. Když se přechází na obrazovku pro přidání či odebrání uživatelů, pošle se seznam uživatelů, kteří již jsou k projektu připojeni. Ten se poté vypíše na řádek, který je vytvořen formou elementu `MultiAutoCompleteTextView`, sloužící pro nápovědy textu. Oproti `AutoCompleteTextView` je možné vyhledávat a přidávat více lidí za sebou na jeden řádek. Po dokončení úprav je zpět poslán upravený seznam uživatelů a také seznam uživatelů, kteří mají být smazáni. Oběma směry je posílán název projektu. Je to proto, že když uživatel změní název projektu a teprve pak chce upravovat uživatele, tak aby po jeho návratu nebyl název projektu přepsán svým původním názvem z databáze.

Seznamy položek jsem vytvářel za pomoci `RecyclerView`, což je pokročilejší a snadnější forma implementace, než pomocí `ListView`. Tuto formu jsem využil u seznamu projektů, reportů a uživatelů na projektu. Pro každý seznam jsem vytvořil zvláštní třídu, která dědí z `RecyclerView.Adapter`. Pomocí tohoto adaptéru se podle mého názoru dají lépe spravovat a implementovat jednotlivé položky seznamu, než jak tomu je u `ListView`.

Pro dosažení implementace Material Designu pro textová pole, ve kterých je textová nápověda nejprve zobrazena na řádku a po zadání znaku přesunuta nad vepsaný text, byl každý `EditText` vložen do `TextInputLayout`. Pro vložení své vlastní horní lišty jsem musel v `Android Manifest.xml` zakázat pro konkrétní obrazovky zobrazení původní lišty. Do své lišty jsem vložil tlačítko do pravého rohu, které je potom v kódu pro konkrétní obrazovku pojmenované podle účelu jaký splňuje (Create, Add).

## Detailní rozložení prvků na hlavní obrazovce

Vykreslení grafu bylo provedeno za pomoci knihovny `MPAndroidChart`<sup>3</sup>, která nabízí spoustu možností, jak graf specifikovat, a podporuje gesta pro posun grafu nebo pro jeho přiblížení a oddálení. Nabízí také možnost vykreslit každou část křivky jiným způsobem, což jsem využil pro odlišení nezadaných reportů. Pro zobrazení ikon uživatelů jsem použil knihovnu `CircularImageView`<sup>4</sup>. Původně jsem chtěl použít některou knihovnu na vytvoření kulatého tlačítka, ale žádné z nich mi neumožnilo správně do něj vložit obrázek, který by se tlačít-

<sup>3</sup>MPAndroidChart — <https://github.com/PhilJay/MPAndroidChart>

<sup>4</sup>CircularImageView knihovna — <https://github.com/Pkmmte/CircularImageView>

kem zakulatil. Ikony nejprve byly vloženy napevno v layoutu, ale když bylo na projektu více uživatelů jak 4, tak se ne všechny ikony vešly na obrazovku. Proto jsem je umístil do `HorizontalScrollView`.

Veškerá tlačítka i uživatelské obrázky na hlavní obrazovce mají pevnou velikost, kterou jsem stanovil na základě testování na různě velikých obrazovkách od úhlopříčky 3,7 až po 5,2. Díky tomu není třeba upravovat velikost obrázků podle displaye nebo dokonce mít pro každý rozlišení zvláštní obrázek. Také jsou pevně přimknuty k rámečkům displaye, čímž je zaručeno, že v závislosti na úhlopříčce se mění pouze velikost grafu. Pro implementaci změny rozložení hlavní obrazovky, když je mobil otočen naležato, jsem vytvořil zvláštní layout, do kterého jsem umístil pouze graf. V kódu pak bylo nutné ošetřit, kdy je obrazovka naležato, aby se nepřístupovalo k elementům, které nejsou k dispozici.

## Funkce NudgeMe

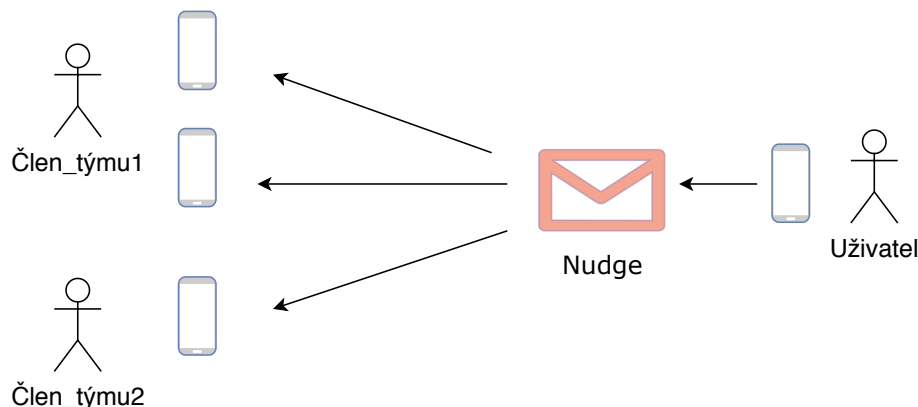
Jak již bylo zmíněno, funkci NudgeMe jsem implementoval jako odesílání zprávy (Nudge) na téma, které je určeno jedinečným identifikátorem uživatele. Nudge je tedy poslán na všechna zařízení, která jsou k danému tématu přihlášena a jejichž uživatelé pro aktuální den nevyplnili report. Toto posílání by se podle mého názoru nedalo provést posláním zprávy na konkrétní zařízení pomocí tokenu příjemce, protože by se těžko určovalo, zda zařízení má o tuto zprávu zájem. Součástí odeslaného Nudgee, jehož struktura je ukázána na listingu 6.1, je i identifikátor projektu, kterého se notifikace týká. Při vytváření notifikace jako reakce na přijatou zprávu, je vytvořen Intent pro přejítí na obrazovku pro vyplnění reportu, kterému je předáno ID projektu. Na vyplňování reportu se nic nemění, ale nebude se report ukládat pro aktivní projekt uživatele, jako když jde uživatel na report z obrazovky grafu, nýbrž pro projekt, který byl poslán přes Nudge.

Když přijde notifikace a uživatel ji otevře na odemčené obrazovce, po kliknutí na ni se otevře okno pro zadání reportu. Problém však nastával při zpracování notifikace na uzamčené obrazovce. Po jejím odemčení se aplikace spustila na hlavní obrazovce. Pro ošetření tohoto případu jsem musel upravit `AndroidManifest.xml`. Pro aktivitu reportu jsem přidal prvek `intent-filter`, ve kterém jsem specifikoval jméno akce a jméno kategorie Intentu. Do Nudgee, který se posílá přes FCM, jsem vložil parametr `click_action`, kterému jsem dal stejnou hodnotu, jakou jsem zadal pro jméno akce v prvku `intent-filter`. Díky tomuto propojení se po kliknutí na notifikaci na zamčené obrazovce uživatel přesměruje přímo na aktivitu projektu.

```
"message":{
  "to" : "/topics/" + userID,
  "notification" : {
    "body" : "PLEASE SEND YOUR REPORT",
    "title" : "Project:" + projectName,
    "priority" : "high",
    "click_action" : "OPEN_REPORT"
  },
  "data" : {
    "projectID" : projectID
  }
}
```

Listing 6.1: Struktura Nudgee, který je poslán pro určité téma, má nastaven nadpis, krátký text, prioritu doručení, jaká akce se má provést po kliknutí a identifikátor projektu.





Obrázek 6.2: Obrázek znázorňující princip posílání Nudgee na zařízení jednotlivých uživatelů, kde notifikací dostane daný uživatel na každém svém zařízení, na kterém je přihlášen k odběru.

## Manipulace s obrázky za pomoci knihovny Glide

Obrázky, které si uživatel nahraje, jsou uloženy ve Firebase Storage a jsou pojmenovány podle identifikátoru uživatele. Změněním obrázku v aplikaci se původní obrázek v databázi nahradí aktuálně vloženým. Před uložením se provede komprese se snížením kvality o 50 % z důvodu snížení jeho velikosti a následněm rychlejšího stahování z úložiště. Načítání pak probíhá pomocí knihovny Glide, která dobře spolupracuje s Firebase. Frekvence stahování obrázků se v aplikaci liší. V nastavení uživatele, kde je možnost si obrázek změnit, se obrázek stahuje pokaždé, kdy uživatel přejde na tuto obrazovku. Je to proto, aby vždy byla viditelná aktuální verze. Na ostatních obrazovkách se obrázky aktualizují pouze jednou denně, poté se stahují z vyrovnávací paměti. Tím se sníží doba načítání a velikost přenesených dat na obrazovkách, které jsou frekventované a na kterých se nachází často více obrázků. V případě, že se obrázek z databáze nepovede z nějakého důvodu načíst, je vykreslen pro uživatele přednastavený obrázek. Při iniciaci Glide jsem vkládal `getApplicationContext()` místo kontextu konkrétní obrazovky. Při použití první varianty v případě opuštění obrazovky dříve, než dojde ke stažení souboru, vznikne chyba a stažení se přeruší, zatímco s kontextem aplikace bude stahování dokončeno na pozadí aplikace. Bezpečnostní oprávnění, která jsem nastavil pro čtení a zápis souborů do úložiště, jsou stejná, jak je ukázáno v listingu 4.3 při popisu Firebase Storage.

Jak již bylo zmíněno, velikost a kvalita obrázků se před uložením sníží kompresí a obrázek je uložen ve formátu JFIF<sup>5</sup>. Snížení kvality je však doposud pevně dané a nezohledňuje velikost vloženého obrázku. Pokud uživatelé vkládají obrázek ze své galerie fotografií z telefonu nedochází k žádným problémům, ale jelikož je možné vložit obrázek i z Google Drive úložiště, může se i po kompresi uložit soubor, který sníží načítání a celkový chod aplikace. I tímto přístupem však dochází k občasnému zpomalení načítání obrázků, než bylo dříve. V předešlé verzi, kdy každý uživatel měl pevně přidělen určitý obrázek avatara, nebylo úložiště využíváno. Všechny obrázky byly uloženy v XML souborech a do databáze se k uživateli uložil pouze název tohoto souboru. Nedocházelo tak k žádnému stahování souboru, ale jen části databáze a stahování bylo v reálném čase.

<sup>5</sup>JFIF — JPEG File Interchange Format

## Práce s databází

Jak již bylo zmíněno v popisu Firebase, číst data z databáze lze několika způsoby. Já jsem využil možnosti číst data jak při jejich změně, tak i jednorázově ve chvíli, kdy funkci volám. Aktualizaci dat při změně jsem použil pro vykreslování grafu a při změně projektu. Tyto informace musí být vždy aktuální a v případě vykreslení aktuálních hodnot v grafu se jedná o jednu ze stěžejních vlastností této aplikace. V ostatních případech se data načítají jen na začátku aktivity. Může se tedy stát, že v okamžiku, kdy je uživatel na určité aktivitě a ve stejnou dobu dojde ke změně dat souvisejících s touto aktivitou, nebudou tyto změny ihned provedeny, ale domnívám se, že aktuálnost těchto dat není nezbytná. Ukládání dat jsem provedl pomocí funkce `updateChildren()`, která seskupí veškeré změny, které se mají provést a provede je všechny současně. Tato metoda je atomická, tedy buď jsou všechna data změněna, nebo žádná z nich. Tím je částečně zabráněno v nekonzistenci dat v databázi. Do aplikace jsem implementoval i podporu pro uchování dat do vyrovnávací paměti. To jsem nastavil pomocí `setPersistenceEnabled(true)`. Tento řádek kódu je třeba umístit před jakoukoliv manipulaci s databází, a proto jsem jej vložil do metody `onStart()`. Aby byla v cache paměti vždy aktuální data, je třeba zavolat funkci `keepSynced(true)` nad odkazem do části databáze, která má být udržována jako aktuální.

## 6.3 Webová aplikace

Pro implementaci jsem použil Javascript, Firebase, Bootstrap a jQuery. Jak již bylo zmíněno v sekci 5.2, implementoval jsem dvě stránky, první pro představení aplikace s krátkým popisem a odkazem na stažení přes Google Play, druhou pro zobrazení všech svých projektů pro přihlášeného uživatele. Každý projekt je zobrazen v elementu `canvas` pomocí knihovny Chart.js<sup>6</sup>. Jedná se o jednoduchou knihovnu, která však oproti MPAndroidChart složitějším způsobem umožňuje definovat křivky různým způsobem, a proto jsem od toho prozatím upustil a všechny křivky jsem vykreslil stejným stylem. Po kliknutí na `canvas` se zobrazí graf přes celou šířku obrazovky. Popisky dat jsou pro zaručení určité přehlednosti vykreslovány nepravidelně každý druhý den. Pokud je graf ve zmenšené variantě a obsahuje hodně dat, tak dochází k častějšímu vynechání dat, které je provedeno automaticky knihovnou Chart.js.

---

<sup>6</sup>Chart.js — <http://www.chartjs.org/>

## Kapitola 7

# Testování

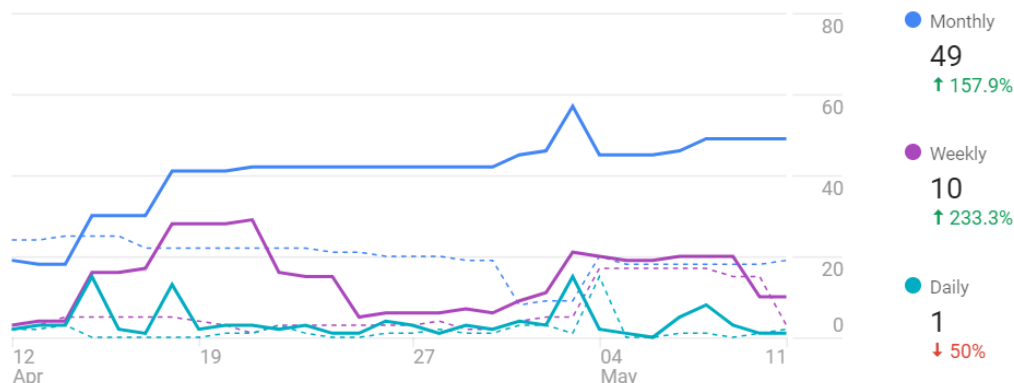
Testování nejprve probíhalo na emulátoru Android Studio a poté na reálných přístrojích. Aby se aplikace správně zobrazovala na různě velikých obrazovkách, bylo třeba upravit design a velikost některých prvků přizpůsobit konkrétní obrazovce. První testování probíhalo v rámci předmětu Tvorba aplikací pro mobilní zařízení, kde jsme v tříčlenném týmu aplikaci používali pravidelně po dobu 2 měsíců. V rámci toho jsme ladili a zkoumali převážně funkcionality jakožto uživatelé, kteří se v aplikaci dobře orientují. Po vytvoření základní funkcionality (posílání Nudgee, zadávání reportů a vytváření projektu) a designu jsem aplikaci vydal jako uzavřenou alfa verzi v Google Play. Do testování jsem zapojil další dva uživatele, kteří aplikaci testovali bez jakékoliv předchozí ukázky funkcionality, pouze na základě popisu chování. Po jejich otestování a následném opravení chyb jsem aplikaci uvolnil jako otevřenou beta verzi. Pro její rozšíření jsem použil Facebook a oslovil několik studentů na Fakultě informatiky Vysokého učení technického v Brně.

Testování probíhalo převážně mezi mladými lidmi, jelikož na ně aplikace cílí. V konečné fázi se do testování zapojilo 10 aktivních uživatelů. V první verzi se uživatelé při prvním spuštění dostali na hlavní obrazovku. Jelikož však zatím neměli vytvořený žádný projekt, tak byl graf prázdný. Z toho důvodu byli občas zmatení a ihned zkoušeli tlačítka pro NudgeMe a pro přechod na vytvoření reportu. Nenapadlo je, že je nutné si nejprve vytvořit projekt a teprve poté je možné používat tyto funkce. Na základě toho jsem aplikaci upravil, aby první uživatelé byli ihned přesměrováni na obrazovku pro vytvoření projektu. Někteří při přidání uživatelů k projektu smazali svůj přednastavený email a tím pádem se jim projekt nezobrazoval v jejich nabídce. Z důvodu toho, aby neprovedli nějakou akci, která by narušila běh aplikace, jsem implementoval různá omezení, jako je znemožnění posílání reportu bez vyplněné zprávy. V reakci na nevyžádanou událost se na displayi zobrazí upozornění.

Typ zařízení	Verze API	Rozměr obrazovky
Emulátor Nexus One	23	3,7
Emulátor Nexus 5	26	4,6
Sony Ericsson ARC S	23	4,2
Samsung Galaxy S6	24	5,2
HTC One M8	21	5

Tabulka 7.1: Seznam zařízení, jejich verze API a jejich rozměry obrazovek, na kterých probíhalo testování aplikace.

Active users ?



Obrázek 7.1: Graf využívání aplikace. Denně aplikaci používá jen jeden uživatel a alespoň jednou týdně kolem 10 uživatelů.

Po testování jsem také odstranil funkci, kdy uživatel v seznamu projektů mohl přejetím prstem po obrazovce zprava doleva zobrazit tlačítko pro smazání a kliknutím na něj projekt smazat. Uživatelé totiž často bez upozornění na tuto možnost nedokázali přijít. Usoudil jsem také, že tato funkcionalita je spíše vhodná pro jiné situace, jako je např. smazání emailů, kdy v seznamu je mnoho položek a uživatel je chce co nejrychleji odstranit. Domnívám se, že v seznamu projektů nedochází k tak rozsáhlým a častým změnám, a proto smazání projektu je možné pouze po rozkliknutí konkrétního projektu.

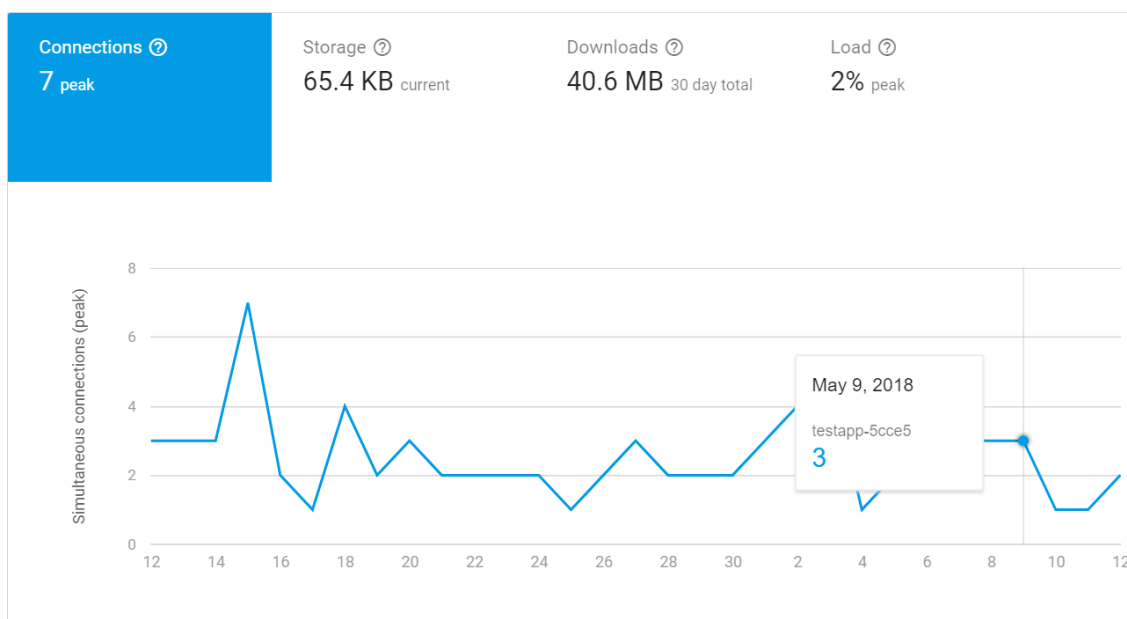
## 7.1 Využití aplikace

Díky využívání Google Play a Firebase jsem dostal statistiky využívání jednotlivých služeb a aktivity uživatelů. Na obrázku 7.1 je možné vidět, jak často je aplikace využívána v čase. Je patrné, že pravidelně ji využívám pouze já sám a ostatní uživatelé jen některé dny. Pravidelnost jsem dokázal zajistit jen upomínáním ostatních pomocí NudgeMe, jinak ostatní z vlastní iniciativy příliš aktivní nebyli.

Jak je možné vidět na obrázku 7.2, tak databáze aktuálně<sup>1</sup> obsahuje kolem 65 kB dat. V těch je aktuálně uloženo kolem 60 projektů a kolem 30 uživatelů (jak skutečných, tak i uměle vytvořených). Zatím se nejedná o závažný problém, ale pokud by počet uživatelů a frekvence používání aplikace narůstaly, bylo by třeba strukturu databáze upravit. V několika případech aplikace potřebuje emailové adresy všech uživatelů a tím v současném strukturování dat stahuje všechny jejich údaje. Počet stažených dat celkově je na 40 MB, přičemž nejvíce za toto zobrazené období bylo v jeden staženo 6 MB.

V úložišti se nachází 17 souborů, které celkově zabírají 1,6 MB dat. Jak je možné vidět na obrázku 7.3 na úložiště je posíláno velké množství dotazů, které je třeba omezit. To jsem zajistil větším využíváním zásobovací paměti pro úložiště a omezením stahování aktuálních obrázků na jeden dotaz denně, ale dosud nedošlo k patřičnému otestování, zda tento pří-

<sup>1</sup>Tento údaj platí ke dni 12.5.2018



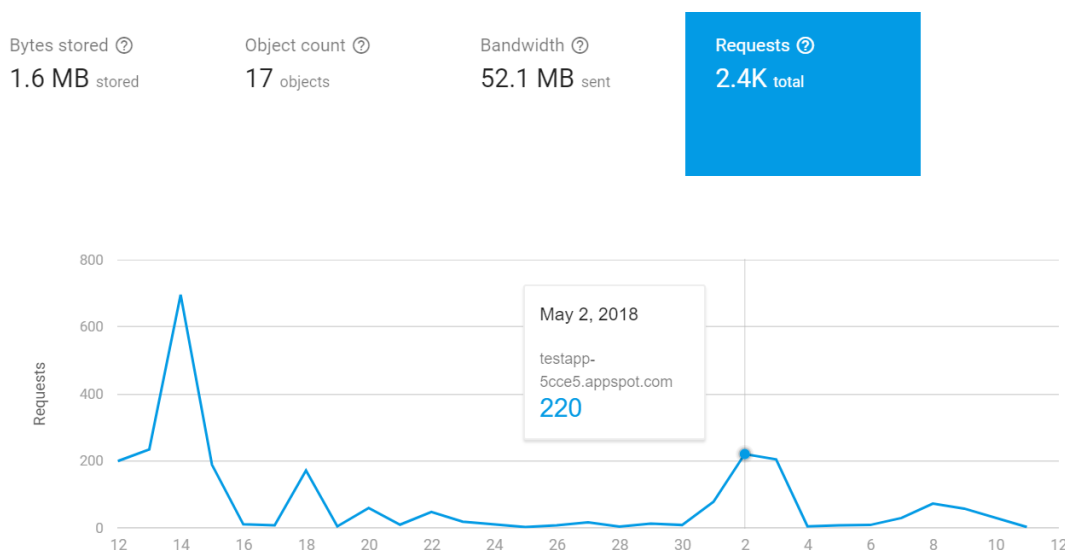
Obrázek 7.2: Zobrazuje graf, který udává, kolik bylo v určitý den v jeden okamžik připojeno uživatelů. Nejvíce jich bylo 15.4.2018, kdy bylo naráz připojeno sedm uživatelů a jinak se jejich počet pohybuje mezi dvěma až čtyřmi uživateli. Dále jsou zde zobrazeny statistiky o celkovém počtu dat v databázi, celkově stažených dat a jaké nejvyšší vytíženosti databáze dosáhla v tomto období.

stup bude výrazně efektivní; graf 7.3 tedy zobrazuje množství požadavků na úložiště před zavedením těchto změn.

Po vydání každé aktualizace byla vždy aplikace otestována na různých typech zařízení jak již bylo zmiňováno v sekci 4.6. U žádné z aktualizací nebyla odhalena chyba, která by vyústila v pád aplikace. Na obrázku 7.4 je možné vidět výsledek tohoto testování pro poslední vydanou verzi. Jak je však patrné z videa vzniklého při testování, umělá inteligence se často nedostala dál, než na hlavní obrazovku. Celková velikost poslední verze APK činí 4,07 MB a je podporovaná na 14963 zařízeních.

## 7.2 Poznatky vyplývající z testování

Po vyzkoušení aplikace někteří uživatelé navrhli, čím by se dala aplikace vylepšit. Nejčastější poznámky se týkaly toho, že není možné zaručit, že uživatel ve svých reportech říká pravdu a hodnotí se objektivně. Dále by se hodilo uživateli při prvním spuštění vysvětlit, jak má postupovat pro vytvoření nového projektu. Přechod od původního designu k Material Designu se setkal s pozitivní odezvou a podle některých působí aplikace více seriózně. Smazáním projektu by také podle některých nemuselo dojít k odstranění dat uživatele z databáze, ale pouze k jeho nezobrazování v seznamu projektů. Do grafu by se mohlo zanést více údajů nebo zlepšit design vykreslených křivek. Přidávání uživatelů na projekt by podle některých mělo být více uživatelsky přívětivé, např. by každý uživatel byl na svém samostatném řádku místo toho, aby byli všichni na jednom. Ustanovení možnosti vložit si vlastní obrázek uživatelé nebrali jako významnou změnu, kterou by potřebovali a klidně by



Obrázek 7.3: Využití úložiště které informuje o počtu uložených dat, počtu uložených souborech, celkovém počtu dat, které uživatelé na úložiště poslali a celkovém počtu dotazů na úložiště.

si vystačili s původními avatary, zvláště když při předchozí variantě byly obrázky rychleji načítány.

Vyplnění reportu uživatelé zvládli bez problému. Málokdy však využili možnosti, zpětně vyplnit report pro den, kdy jej zapomněli odeslat. Tedy nijak je netrápilo, že se jim křivka propadla, i když třeba ten den odvedli kvalitní práci. Co se týče výřečnosti v textu reportu, tak většina lidí popsala svoji práci v jedné větě. Prozatím tedy není třeba implementovat nějaké omezení pro maximální počet zadaných znaků. Někteří by do aplikace implementovali možnost přihlásit se pomocí Facebooku nebo místo emailu používat uživatelské jméno i za předpokladu, že by jej museli vyplnit při registraci. Jedno z možných vylepšení, které by někteří uživatelé ocenili, by mohlo být odeslání notifikace ostatním členům na projektu, informující o tom, že určitý uživatel vyplnil report. Tím by však pozbyla na významu funkce NudgeMe, kdy zprávy by se posílaly trošku neosobně a automaticky. Na druhou stranu tato upozornění by byla méně ofenzivní než Nudge a mohla by ostatní více motivovat díky poukázání na to, že někdo z týmu už se pochlubil svoji prací.

## 7.3 Zhodnocení testování

Z výsledků testování vyplývá, že aplikace pořád ještě není dostatečně uživatelsky přívětivá a testované osoby byly při prvním použití značně nejisté. Tlačítkům pro poslání Nudgee a pro přejítí na obrazovku pro vytvoření reportu by se měly změnit jejich ikony. V současné době uživatelé z jejich podoby těžko chápou, jaká je jejich funkcionality, a proto by měla být nahrazena jinými s lepší popisnou hodnotou. Aplikace byla nainstalována celkově na dvanácti zařízeních, k čemuž došlo při vydání první aktualizace, ale postupně se aktivita uživatelů snižovala až na tři uživatele, kteří ji používají nepravidelně. Tato neaktivita je způsobena tím, že aplikace uživatele na první pohled neohromila natolik, aby se k ní pravidelně vraceli. Dalším důvodem může být, že jsem cílil na špatnou skupinu lidí, kteří

✓	Moto G4 Play ⓘ	9,02 %	548	6,2 tis.	23 mil.	1,3 tis.	▼
✓	P8 Lite ⓘ	3,57 %	812	4,3 tis.	25 mil.	1,8 tis.	▼
✓	Mate 9 ⓘ	2,33 %	668	6,7 tis.	32 mil.	470	▼
✓	Galaxy S7 Edge ⓘ	2,19 %	602	3,2 tis.	81 mil.	2,2 tis.	▼
✓	LG G6 ⓘ	4,08 %	1 tis.	4 tis.	48 mil.	1,1 tis.	▼
✓	Pixel ⓘ	4,03 %	478	5,2 tis.	24 mil.	577	▼
✓	Pixel ⓘ	1,43 %	245	2,2 tis.	9,3 mil.	571	▼
✓	Xperia XZ Premium ⓘ	0,21 %	148	1,3 tis.	4,2 mil.	457	▼
✓	Galaxy J7(2016) ⓘ	1,32 %	232	745	44 mil.	840	▼
✓	Galaxy J1 Ace ⓘ	2,92 %	389	1,3 tis.	21 mil.	-	▼

Obrázek 7.4: Výsledky testování poslední vydané aktualizace na desíti telefonech, které zhodnocují náročnost provedených operací na výkon zařízení. Mezi ukazatele patří průměrné využití procesoru, počet odeslaných a přijatých dat, využití paměti a doba spuštění aplikace.

nejdou nijak soutěživí a nepotřebují se porovnávat s ostatními. Spolu se snadnějším posláním reportu přímo z notifikace a tím i snížením času jeho vyplňování, by uživatelé mohli být více ochotni aplikaci věnovat kus svého času.

Z hlediska fungování aplikace je nutné zlepšit načítání obrázků, aby bylo plynulejší a neomezovalo tak uživatele v používání aplikace, kdy např. při testování v Google Play na mobilu Sony Xperia XZ Premium 13% snímků trvalo vykreslování déle jak 16ms, což značí jisté problémy. Pro případné zvýšení počtu dat v databázi bude třeba jejich strukturování upravit. V rámci testovacích zařízení aplikace působila svižně a kromě načítání obrázků, které je závislé na rychlosti internetového připojení, nevyvstal žádný závažný problém. Aplikace byla vytvořena pro verzi API 25, kromě ní však funguje i pro verzi 21. Pro všechna zařízení je však nutné aktualizovat Google Play Services na nejnovější verzi, bez čehož se nelze do aplikace přihlásit.

## Kapitola 8

# Závěr

Aplikace nedokázala naplnit potenciál, který podle mého názoru měla. Jak již bylo zmíněno ve zhodnocení testování, mohlo k tomu dojít z různých důvodů. Uživatelé nebyli úplně neteční, ale aktivní byli pouze až po přijetí Nudge zprávy. Je možné, že tento přístup je ze začátku nutný a každá skupina bude potřebovat alespoň jednoho člena, který by tímto způsobem donutil ostatní k aktivitě.

V budoucnu se implementují funkce a vlastnosti, které jsou sepsány v sekci 5.4 o návrhu budoucí práce. Taky bude otestována webová aplikace na uživateliích, což jsem zatím nestihl. Dále bude třeba zapracovat poznatky z testování a připomínky uživatelů. Pro případ dalšího testování bude nutné vyhledat jiné skupiny uživatelů, mezi kterými bude více soutěživých typů nebo alespoň lidí, kteří mají ochotu se zlepšit a chtějí pro to něco udělat. Pro zlepšení designu aplikace z hlediska jednoduchosti a přehlednosti ovládání ji otestuji na uživateliích z různých věkových skupin, protože kdyby s jejím ovládáním neměli problém ani starší lidé, tak byla větší jistota, že jej nebudou mít ani mladí. Z uživatelského použití za dobu testování vyplynulo, že stejně často jako pro podporu práce v týmu ji uživatelé využívají pro poměrování se v určité aktivitě. Možná by nebylo špatné otestovat, zda aplikace není vhodná spíše k tomuto účelu. Chtěl bych aplikaci udržet v chodu a vyzkoušet si implementovat některá z navrhovaných vylepšení, zda by alespoň trochu nepřispěla k jejímu zlepšení. V konečné fázi aplikaci uvolním do produkční verze, aby si ji mohl stáhnout jakýkoliv uživatel.



# Literatura

- [1] *Android dokumentace*. [Online; navštíveno 18.05.2018].  
URL <https://developer.android.com/guide>
- [2] *Dokumentace k podepsání APK*. [Online; navštíveno 18.05.2018].  
URL <https://developer.android.com/studio/publish/app-signing>
- [3] *Firebase databáze*. [Online; navštíveno 18.05.2018].  
URL <https://firebase.google.com/docs/database/>
- [4] *Firebase dokumentace*. [Online; navštíveno 18.05.2018].  
URL <https://firebase.google.com/docs/guides/>
- [5] *Firebase uložiště*. [Online; navštíveno 18.05.2018].  
URL <https://firebase.google.com/docs/storage/>
- [6] *Google Play zásady pro vývojáře*. [Online; navštíveno 18.05.2018].  
URL <https://play.google.com/intl/cs/about/developer-content-policy-print/>
- [7] *Material Design*. [Online; navštíveno 18.05.2018].  
URL <https://material.io/>
- [8] Dawson, A.: *Výjimečný webdesign: jak tvořit osobité, přitažlivé, použitelné weby*. Computer Press, Brno, 2012, ISBN 978-80-251-3719-2.
- [9] Fowler, M.; Scott, K.: *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2000, ISBN 0-201-65783-X.
- [10] Franken, R. E.; Brown, D. J.: Why do people like competition? The motivation for winning, putting forth effort, improving one's performance, performing well, being instrumental, and expressing forceful/aggressive behavior. *Personality and Individual Differences*, ročník 19, č. 2, 1995: s. 175 – 184, ISSN 0191-8869.
- [11] Krug, S.: *Web design: nenutte uživatele přemýšlet!* Computer Press, Brno, 2006, ISBN 80-251-1291-8.
- [12] Ludwig, P.: *Konec prokrastinace*. Jan Melvil, Brno, 2013, ISBN 978-80-87270-51-6.
- [13] Rigopoulos, T.: *The Psychology Principles Every UI/UX Designer Needs to Know*. 2016, [Online; navštíveno 18.05.2018].  
URL <https://blog.marvelapp.com/psychology-principles-every-uiux-designer-needs-know/>

- [14] Rivero, J. M.; Grigera, J.; Rossi, G.; aj.: Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. *Information and Software Technology*, ročník 56, č. 6, 2014: s. 670 – 687, ISSN 0950-5849.

## Příloha A

# Obsah přiloženého paměťového média

- /aplikace/mobile/\* — zdrojové kódy pro mobilní aplikaci
- /aplikace/web/\* — zdrojové kódy webové aplikace
- /xpolan07-NudgeMe.mp4 — ukázkové video funkcionality
- /text/\* — zdrojové kódy textu práce
- /xpolan07-NudgeMe.pdf — text práce
- /dokumentace/\* — dokumentace k aplikacím
- /README — informace pro spuštění